UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Planeamento de Transporte a Pedido: Gestão da Procura

Nuno Duarte Dinis e Sousa

Mestrado em Informática

Versão Pública

Trabalho de Projeto orientado por:
Prof.ª Doutora Maria da Graça de Figueiredo Rodrigues Gaspar

2020

Agradecimentos

Convicto de que este documento representa muito mais do que uma dissertação, tenho que agradecer a todos os que fizeram parte desta caminhada que considero ter começado há alguns anos e não apenas este ano.

Antes de mais, dedico este importante marco da minha vida ao meu avô Cristiano Cruz Diniz, que sempre me recebia com um sorriso nos lábios e uma alegria enorme, que sempre me dizia para me agarrar aos estudos para um dia conseguir vingar na vida, e do qual tive de me despedir durante este ano.

Começo por agradecer, como não podia deixar de ser, aos meus pais e à restante família que sempre me apoiaram ao longo de todos os anos, aconselhando-me sempre que necessário.

De seguida, agradecer à Faculdade de Ciências da Unversidade de Lisboa e ao grande DI. A todos os meus colegas e amigos que me acompnharam nestes 6 anos de estudos académicos, nomeadamente aos 14/15, à Cristina Rocha e ao Gonçalo Lobo, que tiveram um papel importantíssimo no início da minha vida académica.

Agradecer aos meus orientadores Professora Doutora Maria da Graça de Figueiredo Rodrigues Gaspar e ao Eng. João Almeida, que se mostraram sempre disponíveis para me apoiar durante a execução deste trabalho de projeto.

Agradeço também a toda a equipa da empresa onde passei este meu útlimo ano, a Card4B, nomeadamente ao João Zenha, que tantas vezes o solicitei, mas que me ajudou sempre da melhor vontade.

Agradeço a Robert Baden-Powell por ter criado o movimento que tanto trouxe à minha vida, desde os ensinsamentos até às fortes amizades que estabeleci através do movimento escutista.

Agradeço a todas as adversidades e a todas as pedras que fui encontrando pelo caminho, pois foi graças a elas que pude crescer ao longo de todos estes anos e permitiram ser o Homem que sou hoje.

At last but not the least, agradeço a Deus pelo dom da vida, por ter permitido que eu chegasse até aqui, nunca desistindo de lutar pelos meus objetivos.

Aos que me acompanharam e me apoiaram nesta caminhada.

Resumo

Este relatório foi redigido no âmbito da disciplina de Dissertação/Projeto C (Informática) do 2º ano do Mestrado em Informática, lecionado na Faculdade de Ciências da Universidade de Lisboa, referindo-se ao trabalho desenvolvido na empresa Card4B - S.A. Systems no contexto do projeto *Planeamento de Transporte a Pedido: Gestão da Procura*, projeto esse que tem como objetivo criar novos mecanismos e melhorar os já existentes que sirvam de base à implementação de um sistema de transporte a pedido.

São várias as razões que levaram as transportadoras de passageiros, e não só, a repensar a forma como efetuam o transporte público de passageiros (e de mercadorias, mas neste projeto o foco são as pessoas): o crescente congestionamento das estradas, a preocupação com as emissões de gases poluentes, a inexistência de transporte de passageiros nas zonas menos populosas, etc. Além das razões operacionais mais evidentes, a grande evolução das soluções tecnológicas abre horizontes a outras soluções de transporte público de pessoas. Neste sentido, surgiu o conceito de *Transport-on-Demand* (ToD). Este conceito pode também ter a denominação de DRT (*Demand-Responsive-Transport*). Uma grande diferença entre o ToD e o transporte público tradicional de passageiros é que o primeiro foca-se no movimento das pessoas e não no movimento de viaturas.

Um outro conceito semelhante, mas diferente do ToD, é a mobilidade como um serviço, comumente abreviado para MaaS (*mobility-as-a-service*). Tal como a evolução tecnológica tem vindo a transformar os serviços de transporte de passageiros e mercadorias, veio também dar azo ao surgimento de novas formas de aceder à mobilidade, facilitando a sua procura e o seu uso. Podemos resumir o MaaS como uma agregação de várias formas de transporte num único serviço de mobilidade acessível a pedido, desde o acesso a todos os serviços individuais de mobilidade até ao fornecimento de informação ao utilizador final sobre informações relevantes de última hora sobre o serviço (atrasos, alterações de rota, etc.), passando pelo planeamento da viagem, marcação e pagamento.

Este trabalho de projeto visou melhorar algumas funcionalidades já existentes em diversos sistemas da empresa Card4B - S. A. Systems, nomeadamente funcionalidades referentes à criação de serviços ocasionais, bem como a visualização de informação de gestão dos serviços de transporte regular. Foram ainda implementadas novas funcionalidades, das quais se destacam a exportação de serviços ocasionais entre sistemas e a sua execução numa aplicação android que é utilizada por motoristas; a implementação da

funcionalidade de unificar diversos serviços ocasionais mediante diversos detalhes a ter em atenção; e ainda, entre outros, o cálculo de diversas informações baseadas nos dados enviados pelas viaturas para o sistema de monitorização, tais como: limites de velocidade excedidos; desvios de rota em relação ao trajeto previsto; tempos médios de paragem por cada paragem do operador em questão.

Palavras-chave: Transporte a pedido; Mobilidade como um serviço; Mobilidade; Flexibilidade; Transporte público.

Abstract

This report was written under the Dissertation / Project C (Informatics) course of the 2nd year of the Master in Informatics, taught at the Faculty of Sciences of the University of Lisbon, referring to the work developed in the project *Transport-on-Demand (ToD) Planner: Demand Management*, a project that aims to create new mechanisms and improve existing ones that will serve as a basis for the implementation of a transport system on demand.

There are several reasons that have led passenger carriers, among others, to rethink the way they carry out public passenger transport (and freight, but in this project the focus is on people): increasing road traffic, concern about pollutant gas emissions, no public passenger transport in less populated areas, etc. In addition to the most obvious operational reasons, the rapid evolution of technology solutions opens horizons for other public transport solutions. In this sense, the concept of Transport-on-Demand (ToD) emerged. This concept may also be called DRT (Demand-Responsive-Transport). One big difference between ToD and traditional public passenger transport is that ToD focuses on people's movement rather than vehicle movement.

Another concept, different from ToD but with similarities to it, is mobility-as-a-service, commonly abbreviated to MaaS. As technological developments have been transforming passenger and freight transport services, it has also given rise to the emergence of new ways of accessing mobility, facilitating their search and use. We can summarize MaaS as an aggregation of multiple forms of transport into a single on-demand mobility service, from access to all individual mobility services to providing the end user with relevant last-minute service information (delays, route changes, etc.), including trip planning, booking, and payment.

This project work aimed to improve some functionalities already existing in several systems of the company Card4B - S. A. Systems, namely functionalities related to the creation of occasional services, as well as the visualization of management information of the regular transport services. New features have also been implemented, of which stand out: the export of occasional services between systems and their execution in an android application that is used by drivers; the implementation of the functionality to unify several occasional services through several details to be taken into account; and yet, among others, the calculation of various information based on the data sent by the

vehicles to the monitoring system, such as: speed limits exceeded; deviations from the route in relation to the planned route; average stop times for each stop of the operator in question.

Keywords: Transport-on-demand; Mobility as a service; Mobility; Flexibility; Public passenger transport.

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas

API Application Programming Interface. xvii, 13, 16, 23, 24, 27, 28

CSV Comma-Separated Values. 12

JSON JavaScript Object Notation. 42

KML Keyhole Markup Language. 20, 21

MaaS Mobility-as-a-Service. 4, 5, 10

MVC Model-View-Controller. 23

OSM OpenStreetMap. 18

OSRM Open Source Routing Machine. 18, 19, 22, 33

RGB Red, Green, Blue. 37

SML Standard ML. 14

SP Stored Procedure. 24, 26, 36, 39

ToD Transport on Demand. 1, 2, 3, 4, 5, 6, 16, 17

TPA Terminal de Pagamentos Automáticos. 15

URL Uniform Resource Locator. 44, 45

UTC Coordinated Universal Time. 13, 26

WKT Well-Known Text. 20, 39, 40

XML Extensible Markup Language. 14, 20

Capítulo 1

Introdução

São inúmeros os casos em que o surgimento de aplicações móveis melhorou a vida das pessoas. O acesso à informação através de apenas um dispositivo móvel, como o smartphone, facilitou sem qualquer dúvida o dia-a-dia da população. Notícias, previsões meteorológicas, acompanhamento de resultados desportivos, são alguns exemplos. No que concerne à mobilidade, a evolução tecnológica também contribuiu bastante para facilitar o movimento das pessoas. Mas será que já foi tudo feito o que está ao nosso alcance?

É verdade que temos na palma da nossa mão inúmeras ferramentas que, de um modo ou de outro, facilitam a mobilidade das pessoas e outras tarefas do quotidiano: consultar horários de transportes públicos, fazer um pedido de transporte individual e remunerado de passageiros em veículos descaracterizados a partir de plataforma eletrónica (TVDE), fazer pagamentos sem necessitar de dinheiro nem de cartões eletromagnéticos, planear uma rota através de aplicações de mapas (Google Maps, Waze, etc.). Mas, e se tivéssemos algo que juntasse isso tudo? É essa a proposta da mobilidade como um serviço, habitualmente designada pela sigla maas (*mobility-as-a-service*) usada daqui em diante neste relatório. O maas resume-se numa plataforma que agrega vários serviços de acesso à mobilidade com um único canal de pagamento, em vez de o passageiro ter de adquirir diversos títulos e proceder a vários pagamentos.

Por outro lado temos o tod (*transport-on-demand*), que é uma estratégia que combina o transporte regular de passageiros rodoviário com os TVDE, ou seja, temos um transporte público de passageiros rodoviário adaptado à procura em que é o cliente quem desencadeia a viagem ao solicitá-la, contrariamente ao que acontece no transporte regular de passageiros mais comum, onde temos viagens regulares pré-definidas que se realizam independentemente do número de pessoas que irão utilizar esses mesmos serviços.

1.1 Motivação

Existem muitas razões que podem levar à implementação de plataformas maas e/ou de estratégias de ToD. Desde logo a mais evidente é aumentar e integrar o acesso a diferentes

formas de mobilidade das pessoas. Isto significa não só melhorar o acesso para quem já é cliente dos serviços de transporte público, mas também atrair quem ainda tem a sua viatura pessoal como principal meio de transporte no dia-a-dia. Isto ajudaria a reduzir o congestionamento nas estradas. Pegando em Lisboa como exemplo, segundo o estudo TomTom Traffic Index 2018 [13], a capital portuguesa é a 77ª cidade com mais trânsito entre as 403 cidades incluídas no estudo. Este estudo indica que quem anda por Lisboa passa 32% de tempo adicional em trânsito (média). Esta é a percentagem média tendo em conta todas as horas do dia, todos os dias da semana, mas nas horas de ponta aumenta: nos dias úteis entre as 08h e as 09h este valor é superior a 65%; entre as 18h e as 19h é superior a 70%. Nas horas de maior congestionamento, os utilizadores das estradas lisboetas levam 20 minutos adicionais no trânsito de manhã e 22 ao fim do dia, tendo em conta uma viagem de 30 minutos, isto é: de manhã um percurso que demoraria 30 minutos em condições normais, na verdade demora 50; ao fim do dia, piora ligeiramente, para 52 minutos. Neste estudo são apresentadas mais 4 cidades portuguesas (onde, recorde-se, a classificação é em relação às 403 cidades analisadas e a percentagem é o tempo adicional que uma pessoa leva a fazer um certo trajeto, tendo em conta uma viagem de 30 minutos): Porto (121º, 28%), Funchal (336º, 16%), Braga (342º, 16%) e Coimbra (371º, 14%). Em todas elas concluiu-se que é fora das vias rápidas que ocorre mais trânsito, algo que é favorável à implementação de estratégias de ToD.

Algo que está ligado aos transportes é a poluição atmosférica. Agora que tanto se fala de crise climática, este é também um ponto a favor da implementação de estratégias que promovam a mobilidade de passageiros através de transportes públicos, uma vez que se o número de pessoas que usam a viatura pessoal como meio de transporte pessoal diminuir, vamos ter menos viaturas a circular e consequentemente menos emissores de poluição atmosférica. O diesel é a principal causa de poluição atmosférica em alguns países desenvolvidos [9], e em Portugal são os carros a diesel que têm sido os mais utilizados nos últimos anos: segundo a PORDATA [11] desde 2010 que a utilização de carros a diesel tem vindo a aumentar consecutivamente, passando de uma representação de 57,4% em 2010 para 65% em 2018. Assim, este também é um ponto fundamental num acesso mais fácil e mais eficiente à mobilidade, sobretudo se esta for realizada com recurso a uso de energias renováveis.

A integração de várias formas de aceder à mobilidade pode também ser uma ferramenta para combater um fenómeno que assola Portugal há vários anos: o despovoamento. Com duas dezenas de concelhos com uma densidade populacional abaixo dos 10 habitantes/km^2 em 2018 [10] e com perspetivas de uma diminuição da população em Portugal a médio prazo [3], são numerosas as aldeias e vilas que, devido ao reduzido número de potenciais utilizadores, não possuem uma rede de transportes regular de passageiros, algo que pode ser combatido com a implementação de serviços ToD. Esta possível solução poderá não só manter a população atualmente residente nos locais com maior risco de des-

povoamento, como pode também aumentar o movimento de pessoas nessas regiões, uma vez que o ToD pode também conter uma componente para o turismo.

1.2 Objetivos

O Projeto *Transport-on-Demand (ToD) Planner for MaaS - Demand Management* tem como objetivo criar um sistema de ToD para transporte público com a possibilidade de combinar serviços, desde um regime pré-definido até à flexibilidade total. A flexibilidade incide em 3 aspetos: Recursos para a execução dos serviços (tipologia dos veículos, motoristas, manutenção, etc.); Procura (pedidos de clientes individuais, alugueres para grupos, etc.); Serviço (itinerário, paragens, horários, etc.).

O sistema tem como objetivo fornecer suporte a funcionalidades tais como:

- Aceitação de pedidos e reserva de serviços em função da disponibilidade de meios;
- Otimização dos itinerários dos serviços com base em motores e APIs abertas (e.g. here.com, Google, Open Street Maps), aplicação de um algoritmo existente, e flexibilidade do workflow no BackOffice;
- Informação aos utilizadores sobre itinerários previstos;
- Início, execução e conclusão dos serviços.

Os objetivos específicos a atingir neste trabalho de mestrado, no âmbito do projeto da Card4B, serão apresentados adiante na secção 3.1, após apresentar a informação sobre a versão atual da estrutura dos sistemas já existentes na Card4B, necessária para a sua compreensão.

1.3 Instituição de Acolhimento

A Card4B - S. A. Systems [2] é uma empresa com 12 anos que surgiu da identificação da necessidade de disponibilizar soluções para uma ”nova cultura de mobilidade”nos centros urbanos, e tem como principal foco o fornecimento de componentes de software e serviços especializados de soluções integradas de mobilidade e *city-services*, como transportes públicos, parques de estacionamento dentro e fora das localidades, portagens, táxis, partilha de carros, partilha de bicicletas, transporte a pedido, escolas, bibliotecas, piscinas, estádios, museus, entre outros. Esta instituição desenvolve e opera soluções integradas de mobilidade através da bilhética interoperável sem contato, informação ao passageiro, sistemas embebidos e smartphones, integração de sistemas e *business intelligence*. A abordagem é feita com base no conceito de *Ticketing Kernel* [1] para o diálogo entre todos

[1] Sistema operativo dos TPAs - terminais de pagamentos automáticos.

os tipos de terminais e mídias (por exemplo, cartões). Este projeto está a ser supervisionado por duas pessoas da Card4B - S.A. Systems: João Miguel Correia da Silva Carreira Almeida e João Filipe Ramos Zenha.

1.4 Estrutura do documento

Este documento está organizado nos seguintes capítulos:

- Capítulo 1 - Do qual faz parte esta secção, contém uma apresentação do conteúdo a abordar neste projeto, os objetivos do projeto, a instituição onde está a ser realizado o projeto e a organização do relatório.
- Capítulo 2 - Descrição de alguns exemplos de trabalhos já feitos na área do ToD e do MaaS.
- Capítulo 3 - Breve análise da arquitetura de cada um dos sistemas que compõem este trabalho, bem como dos principais desafios deste projeto.
- Capítulo 4 - Neste capítulo está descrito todo o trabalho que foi desenvolvido durante o projeto, dividido em várias secções que representam cada fase diferente deste projeto.
- Capítulo 5 - Sumário da conclusão do trabalho desenvolvido e descrição de possível trabalho futuro relacionado com as funcionalidades implementadas neste projeto.

Capítulo 2

Trabalho relacionado

Neste capítulo vão ser apresentados alguns exemplos de projetos dos 2 grandes temas deste projeto: *Transport-on-Demand* e *Mobility-as-a-Service*. Ainda que sejam conceitos algo recentes, já existem (ou existiram) diversos projetos, tanto de ToD como de MaaS. De seguida serão descritos 3 exemplos de ToD e 2 de MaaS, embora existam mais que, eventualmente, poderão ser descritos também no futuro em função das suas características.

2.1 Transport-on-Demand

2.1.1 Kutsupulus

Kustupulus foi um projeto desenvolvido através de uma parceria entre a Autoridade de Transporte Regional de Helsínquia (HSL) e a Split Finland (antiga Ajelo Ltd.), que funcionou entre 2012 e 2015 em Helsínquia, capital da Finlândia. O seu principal objetivo era combater o crescente número de pessoas que utilizavam o seu carro pessoal como meio de transporte, tentando transferi-las para o transporte público.

Conforme consta no relatório final deste projeto [6], a Kutsupulus ”foi um sucesso, tanto tecnologicamente como sob a perspetiva da satisfação do cliente”. Os utilizadores deste serviço podiam solicitar viagens através de uma aplicação móvel, bem como efetuar o pagamento. As viagens através da Kustupulus eram realizadas entre paragens virtuais fixas.

Um dos grandes desafios era calcular as rotas com base nos pedidos efetuados pelos utilizadores, permitindo que os passageiros com percursos próximos uns dos outros pudessem ser transportados na mesma viatura, sendo que para isto entra em jogo um fator (quase) imprevisível: o congestionamento de trânsito. Para isso foi desenvolvido um software com a finalidade de permitir uma rápida adaptação às mudanças da intensidade do trânsito, bem como de possíveis desvios devido a novos pedidos de viagens por parte de outros clientes.

Uma vez que para os passageiros a pontualidade é um dos grandes fatores que mede a qualidade de um serviço, a Kutsuplus decidiu operar apenas entre paragens virtuais

em vez de locais definidos pelos utilizadores, reduzindo assim o número de caminhos possíveis entre cada ponto de recolha/entrega de passageiros, melhorando assim substancialmente a estimativa de tempo de espera por parte dos utilizadores, bem como a estimativa da duração da viagem.

Nos 4 anos de serviço registaram-se mais de trinta mil utilizadores, foram realizadas mais de cem mil viagens e a receita de viagens foi cerca de 900.000 euros.

Apesar do reduzido número de viaturas a operarem, 15, os clientes mostraram-se muito satisfeitos, mas em 2015 a HSL decidiu que a Kutsupulus cessaria funções na sua forma atual (à época).

O forte crescimento da procura ao serviço implicava um aumento do número de viaturas disponíveis para a execução das viagens, garantindo uma melhoria da disponibilidade, qualidade e economia do serviço, mas tal não aconteceu. A HSL chegou a propor aos municípios membros do projeto que a frota fosse triplicada, passando de 15 para 45 veículos, mas tal proposta não foi atendida. Esta recusa prejudicou bastante o desenvolvimento dos serviços.

2.1.2 MOIA

A MOIA é uma marca de mobilidade do grupo Volkswagen que foi anunciada no final de 2016 e tem como missão tornar a mobilidade urbana mais acessível e democrática, oferecendo um conjunto de soluções avançadas no domínio da conetividade e partilha de viaturas.

A Split Finland, mencionada na secção anterior, foi adquirida pela MOIA em meados de 2017 para iniciar o desenvolvimento do serviço de táxi a pedido prevendo reduzir substancialmente custos através de viagens partilhadas numa mesma viatura. Porém, antes desta aquisição, a Split realizou uma solução de ToD em Washington DC.

Replicando o modelo usado em Helsínquia (como referido na secção anterior, resposta a serviços a pedidos com paragens virtuais fixas), a Split estava agora a competir com outros serviços a pedido, como a Uber ou a Lyft. Além destas, a Split pretendia também competir com as soluções de transporte público ali existentes.

A Split, que oferecia viagens desde 3 dólares (2 dólares de tarifa básica e 1 dólar por milha), com 100 motoristas e depois de transportar milhares de dezenas de passageiros, com um aumento médio de passageiros de 30% por mês [12], cessou as suas funções em outubro de 2016. A empresa alegou que o mercado de *ride-hailing* [2] estava saturado. As promoções realizadas pela Uber e pela Lyft representaram um problema para empresas como a Split. Não sendo capaz de cumprir a missão de ser uma solução mais barata e eficiente que o *ride-hailing*, e mais flexível que o transporte público comum, a Split interrompeu o seu serviço.

[2] Serviço em que uma pessoa contrata um motorista para fazer uma viagem desde o ponto onde pretende ser recolhido até ao ponto onde pretende ser entregue (serviço de táxi, Uber, Lyft, etc.).

Ao comprar a Split, a MOIA adquiriu assim o pilar fundamental do conhecimento técnico e os algoritmos pioneiros desenvolvidos anteriormente para um novo modelo de negócios de transporte a pedido.

O modelo de negócio da MOIA assenta num sistema totalmente elétrico e ecológico. Assenta num design inovador de veículo com um maior conforto para os passageiros ao adicionar mais espaço a cada um dos 6 lugares disponíveis por viatura. Cada lugar dispõe de entradas USB e luzes auxiliares de leitura. É também disponibilizado acesso Wi-fi para todos os ocupantes do veículo.

Em suma, este é um serviço que é o resultado de uma experiência de ano e meio em Washington DC, que se concentra em oferecer um serviço de transporte a pedido de alto conforto ao público e uma solução ecológica.

2.1.3 Médio Tejo

A Comunidade Intermunicipal do Médio Tejo (CIMT) é uma comunidade que abrange vários municípios do Centro de Portugal Continental com uma área geográfica de atuação de 3.344/km^2 e integra 13 concelhos, num total de 247.330 habitantes (censos 2011) [7]. A CIMT foi criada no seguimento das extinções da Comunidade Urbana do Médio Tejo e da Associação de Municípios do Médio Tejo e tem o objetivo de promover o desenvolvimento equilibrado e sustentável do seu território de intervenção, com base no planeamento estratégico regional e o apoio às autarquias locais [7]. Uma das áreas de atuação é o transporte.

Devido à existência de zonas no Médio Tejo de povoamento rarefeito e de baixa densidade, torna-se inviável a existência de transporte público regular, uma vez que a procura não compensa os custos da possível oferta. Assim, para assegurar a mobilidade da população nos espaços rurais e promover a inclusão social, a CIMT criou um projeto chamado ”Transporte a Pedido no Médio Tejo”, constituindo assim uma nova solução de transportes que cobre uma área territorial mais ampla, com custos controlados e níveis de serviço mais adequados à realidade local. Por outro lado, existem também casos em que a necessidade de cobrir a região é penalizada por constantes desvios de rota que tornam os circuitos muito demorados, reduzindo bastante a sua eficiência. Nas zonas em que estas situações se verificam faz claramente sentido a existência de um sistema de transporte a pedido.

Este serviço assenta em circuitos pré-definidos, constituídos por paragens estabelecidas pela própria CIMT. Os horários variam dependendo da época do ano (período escolar ou férias escolares), e as viagens só são realizadas caso existam reservas até às 15h do dia anterior. Após as 15h ainda são aceites reservas, mas apenas para paragens que já tenham sido incluídas em reservas anteriores. Ao contrário dos 2 exemplos anteriores de transporte a pedido, neste serviço as reservas não são feitas por meio de uma aplicação móvel, mas sim por chamada telefónica, que é gratuita.

As viagens são realizadas essencialmente por táxis de 4 ou 8 lugares, sendo que em Vila Nova da Barquinha encontra-se também ao serviço uma viatura de transporte escolar. As viagens são cobradas pelo motorista quando um passageiro entra na viatura e os preços vão de 1,20 euros até 3,80 euros, sendo que desde abril de 2019 , no âmbito do Programa de Apoio à Redução Tarifária (Despacho n.º 1234-A/2019 [4]) os escalões tarifários passaram a ter bilhetes desde 1 até 2 euros, resultando num desconto de cerca de 47% no maior escalão tarifário [7].

Este projeto piloto teve início no começo de 2013, no concelho de Mação. Foi escolhido este concelho devido à sua dispersão urbana, à grande falta de transportes regulares e também devido ao envelhecimento geral da população. Em meados de 2014 o serviço foi alargado aos concelhos do Sardoal e de Abrantes. Entre 2016 e 2017 foi alargado aos concelhos de Ourém, Tomar, Vila Nova da Barquinha, Alcanena, Constância, Sertã, Ferreira do Zêzere e Torres Novas, totalizando assim 60 circuitos, mais de 1.200 paragens e cobrindo uma população de cerca de 123.000 habitantes.

Porém, ainda existem regiões onde a oferta de transporte coletivo regular é escassa, prevendo assim um alargamento deste serviço a outras zonas a curto-prazo.

Este serviço contou com o apoio da União Europeia e conta ainda com investimentos do Fundo para o Serviço Público de Transportes. Conta também com o apoio do Fundo Ambiental no âmbito do Programa de Apoio à redução Tarifária.

2.2 Mobility-as-a-Service

2.2.1 UbiGo

A UbiGo é um serviço de maas que está em funcionamento em algumas regiões da capital da Suécia, Estocolmo. Este serviço baseia-se num projeto piloto realizado também na Suécia, mas na cidade de Gotemburgo, em 2013. Desse projeto piloto foi possível concluir, entre muitas outras coisas, que apenas 3% dos participantes preferia continuar a usar a sua viatura pessoal em vez de viajar através da UbiGo [8].

O projeto piloto, que se chamava também UbiGo, foi desenvolvido dentro do projeto *Go: Smart*. Neste projeto, que teve a duração de 6 meses, participaram quase 200 pessoas num total de 83 domicílios. A UbiGo tentou preencher um espaço que existia entre o transporte público e o transporte privado, disponibilizando serviços de transporte personalizados consoante as necessidades de cada passageiro. A solução foi realizada unindo fornecedores de transporte já existentes, incluindo partilha de carros (*car-sharing*) e bicicletas (*bike-sharing*), táxis e aluguer de carros. Todos estes serviços eram agrupados num único serviço de assinatura.

A oferta deste serviço era feita através de uma aplicação móvel que continha diversas funcionalidades: ativar viagens, realizar/verificar reservas, aceder a passagens já ativadas, verificar o saldo, bónus, histórico de viagens e obter suporte (através de perguntas

frequentes ou atendimento ao cliente).

O projeto foi um grande sucesso. No final do projeto, 97% dos participantes queria continuar a utilizar o serviço, enquanto que os restantes 3% preferia voltar a usar a sua viatura pessoal. A satisfação dos clientes foi subindo à medida que o projeto ia avançando, desde 75% (antes do projeto iniciar) até 93% (final do projeto)[8]. Além da satisfação de cada passageiro, é importante também realçar a mudança de atitude em relação aos modos de viagem. À medida que o projeto ia decorrendo, os participantes desligavam-se dos transportes pessoais, passando a preferir o uso de outros modos de mobilidade. Custo, conveniência, flexibilidade e (re)descoberta de modos alternativos foram os grandes fatores que impulsionaram esta mudança de pensamento.

5 anos após este projeto ter terminado, o sistema foi lançado em Estocolmo, no início de 2019. ”As viagens tornar-se-ão mais eficientes, inteligentes e individualizadas, facilitando, em última análise, as pessoas na região de Estocolmo que estão constantemente a lutar para lidar com as complicações da vida quotidiana”, referiu o presidente da Stockholm Lokaltrafik, operadora de transporte público da cidade que é parceira deste projeto. Já o chefe-executivo da UbiGo, Hans Arby, refere que ”Ubigo significa que menos pessoas em Estocolmo precisam de possuir carros, o que reduzirá o congestionamento nas ruas e aumentará as viagens de transporte público. Sem mencionar menos necessidade de lugares de estacionamento”. Por último, Daniel Helldén, comissário de trânsito da cidade de Estocolmo disse: ”O serviço oferece uma maneira flexível e conveniente de viajar com base nas suas necessidades pessoais. Isso aumentará a mobilidade para mais pessoas sem a necessidade de possuírem um carro, beneficiando não apenas o indivíduo, mas também a sociedade e o meio ambiente”[8].

2.2.2 Whim

Whim é o aplicativo de mobilidade como serviço da empresa MaaS Global. A empresa lançou o serviço Whim em Helsínquia no final de outubro de 2016, seguido por um lançamento completo em 2017.

Tal como a UbiGo, a Whim combina várias formas de transporte numa única assinatura e viajar era possível através de uma única aplicação móvel. Através desta aplicação os clientes podem planear a sua viagem, efetuar reservas e os respetivos pagamentos.

Ao fim do primeiro ano de operação foi feito um estudo, Whimpact [1], para perceber qual o impacto deste serviço na comunidade local. Foram realizadas mais de 4 milhões de viagens e até à data do estudo eram cerca de 8.500 os subscritores deste serviço na capital finlandesa. Os utilizadores da Whim realizavam uma média de 2,15 viagens por dia, contrastando com as 1,46 viagens realizadas pelo cidadão comum. De todas as viagens de bicicleta efetuadas, 42% eram combinadas com outros meios de transporte público. As bicicletas e os táxis eram utilizados para viagens de ”primeira/última milha”, isto é, viagens entre a origem/destino e as paragens de outros meios de transporte, revelando

que estes viajantes estão de facto imersos na multimobilidade. Em comparação com os outros cidadãos da cidade, os utilizadores deste serviço combinam 3 vezes mais o táxi com transportes públicos.

Neste estudo [1] foi possível também concluir que a Whim não trouxe apenas benefícios para a viagem de cada utilizador, trouxe também algumas vantagens para o meio ambiente e financeiro: um maior uso de uma gama mais ampla de transportes tende a reduzir as emissões de CO2; quando o uso de transportes públicos aumenta, diminui o congestionamento do trânsito nas cidades, podendo levar também a um maior uso do espaço urbano pelas pessoas; o MaaS pode levar a escolhas mais eficientes, saudáveis e ecológicas.

Capítulo 3

Análise e Desafios

3.1 Análise

Este trabalho é realizado no âmbito de 3 sistemas já existentes desenvolvidos anteriormente na Card4B - Systems S. A.: ***Planeamento***, ***Monitorização*** e ***AppMotorista***. Estes 3 sistemas interagem entre si e também com outros módulos. Esta interação pode ser representada da seguinte forma:

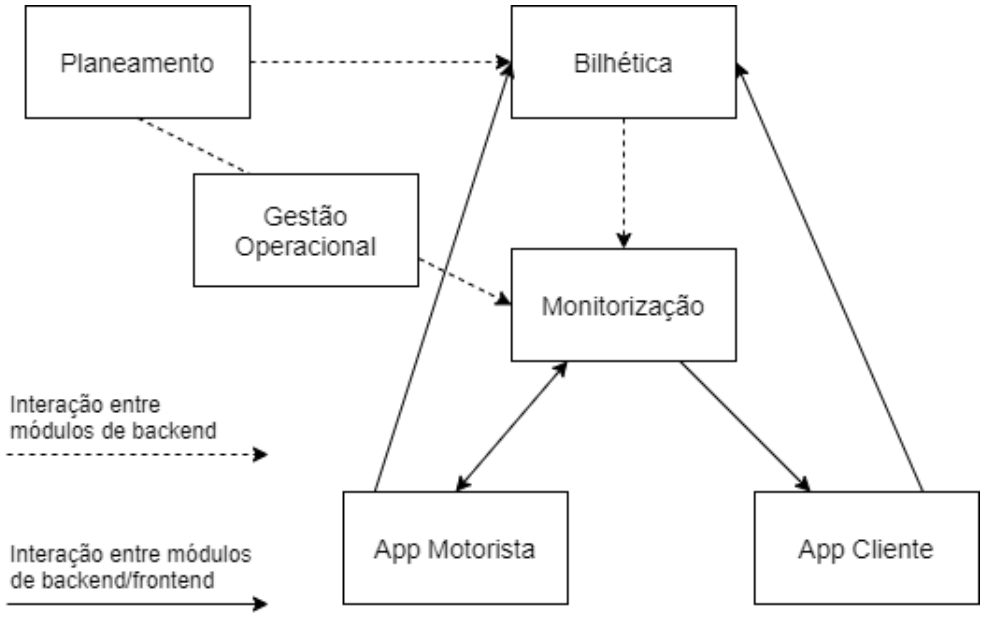

Figura 3.1: Diagrama dos sistemas existentes.

De seguida serão abordados mais pormenorizadamente cada um destes sistemas.

3.1.1 Planeamento

O ***Planeamento*** é o módulo de *BackOffice* (acesso web) de planeamento do serviço de transporte, incluindo serviços, recursos humanos e materiais. Este módulo gere as configurações base para alimentar a ***Monitorização*** e a ***Bilhética***. Pode ainda integrar-se com o Gestão Operacional, direccionado para a expedição, quiosque motorista, e consola de portaria, manutenção, oficina, lavagem e estacionamento, etc.

Os dados do ***Planeamento*** dependem do modelo de negócio de cada empresa, e como tal estes dados provêm de uma fonte externa que consiste num conjunto de ficheiros CSV que serão importados e lidos pelo ***Planeamento***, e o seu conteúdo será posteriormente inserido na base de dados do ***Planeamento***.

O ***Planeamento*** possui uma interface web onde o cliente (operador) pode, sumariamente, consultar, adicionar, alterar e remover informações sobre:

- Unidades
- Viaturas
- Motoristas
- Paragens
- Linhas
- Circulações
- Horários
- Épocas
- Tipos de dia
- Planeamento de operações
- Serviços ocasionais

A base de dados do ***Planeamento*** pode ser populada de 2 formas distintas: através da importação de ficheiros CSV fornecidos pela empresa cliente, como já foi referido acima; e pode ser populada através da interface web do ***Planeamento***, uma vez que através desta o utilizador pode aceder, escrever, alterar ou remover dados da base de dados.

Figura 3.2: Organização do Sistema ***Planeamento***

3.1.2 Monitorização

A ***Monitorização*** é o *BackOffice* de monitorização de veículos, também configurável, que permite seguir a frota em tempo real e alimentar outros módulos, como o ***Planeamento*** ou a ***AppMotorista***, facilitando a otimização da operação de transportes, a disponibilização de informação em tempo real, dos dados de localização e dos horários, ao longo do serviço, das viaturas e motoristas. Estes dados podem ser obtidos através de uma APP ***AppMotorista*** na viatura, ou outros módulos de localização.

À semelhança dos outros módulos, a ***Monitorização*** também possui uma *web interface*, mas ao contrário do ***Planeamento*** e da ***Bilhética*** (abordado mais à frente), esta *web interface* serve apenas para consulta de dados, não estando disponível nenhuma funcionalidade que permita adicionar, alterar ou remover dados da base de dados da ***Monitorização***.

A base de dados da ***Monitorização*** é preenchida através do ***Planeamento*** 1 vez por dia e os dados inseridos são referentes ao próprio dia e ao dia seguinte. Este preenchimento dá-se por volta da meia-noite (em hora UTC, que no caso de Portugal Continental corresponde ao horário de inverno; portanto, no Verão, a atualização dá-se por volta da 01h).

A ligação entre a ***Monitorização*** e a ***AppMotorista*** é realizada por meio de uma API do tipo RESTful, o que significa que podem ser feitos pedidos do tipo GET, HEAD, POST e DELETE (entre outras, mas são estas 4 que estão disponíveis nesta API).

O utilizador, a partir do momento em que faz login no módulo ***Monitorização***, tem acesso a todas as funcionalidades (que foram adquiridas pela sua empresa), não existindo uma hierarquia de utilizadores.

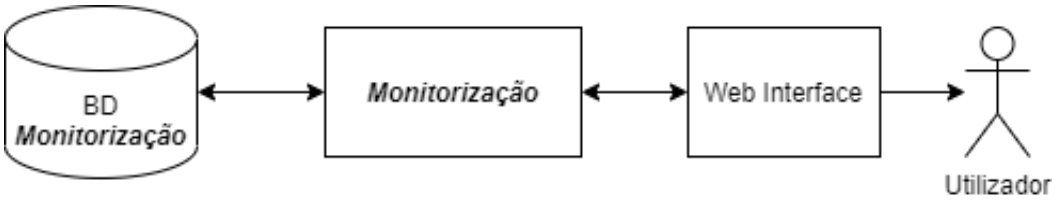

Figura 3.3: Organização do Sistema ***Monitorização***

3.1.3 Bilhética

Apesar de não terem sido feitas alterações neste sistema, foi necessário compreender minimamente como é que funciona devido à ligação entre o sistema ***Monitorização*** e o sistema ***AppMotorista***.

A ***Bilhética*** é o *BackOffice* de gestão da bilhética, assim como o fluxo de dados com os módulos periféricos do sistema ***AppMotorista***. Nesta aplicação é possível configurar a oferta de títulos de transporte, as respetivas regras tarifárias, topologia da rede, o *layout* dos recibos e dos cartões personalizados; permite igualmente personalizar os cartões sem contacto e vender títulos de transporte, controlar todo o fluxo financeiro das vendas, gerir a informação dos clientes, emitir faturação certificada e listas de autorização e exclusão. Tem ainda um mecanismo de listas de recarga e/ou de autorização.

Os dados da ***Bilhética*** vêm diretamente do cliente uma vez que incluem dados de configuração relacionados com bilhética. Estes dados são fornecidos através de um ficheiro XML .

Depois de fornecido o ficheiro XML por parte do utilizador, é utilizado um conversor que irá converter este ficheiro num outro ficheiro que será escrito também em XML e depois convertido num ficheiro SML , para que este seja legível pela ***Bilhética*** de forma a poder ser importado para a base de dados da ***Bilhética*** e para, eventualmente, ser utilizado noutros módulos.

A base de dados da ***Bilhética*** pode ser populada de 2 formas distintas: através de um ficheiro fornecido pela empresa cliente que, após ser convertido num outro ficheiro, é importado para a base de dados (como já foi referido acima); e pode ser também populada através da interface web da ***Bilhética***, já que esta permite ao utilizador adicionar, alterar ou remover dados da base de dados.

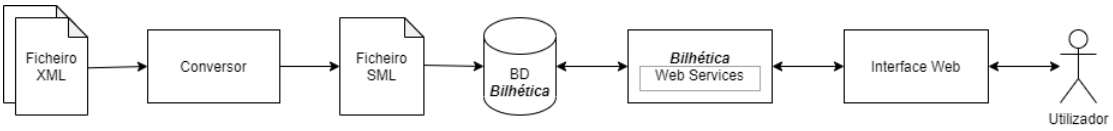

Figura 3.4: Organização do Sistema ***Bilhética***

3.1.4 AppMotorista

A ***AppMotorista*** é um sistema que funciona como *front-end* e trata-se de uma aplicação Android que é instalada num tablet ou smartphone que estará no autocarro e será utilizado pelo motorista deste. Esta aplicação encapsula as funcionalidades de bilhética da ***Bilhética*** e de gestão de serviços da ***Monitorização*** e interage com o cliente final através de um terminal de pagamentos bancários (TPA) que permite: impressão de bilhetes e faturas, leitura/validação de cartões e pagamentos com cartão bancário. A ***AppMotorista*** necessita da ajuda de um módulo dedicado a terminais de pagamentos bancários para funcionar por completo (este módulo não é abordado neste relatório).

A ***AppMotorista*** requer uma sincronização para obter os dados da base de dados da ***Bilhética*** e da base de dados da ***Monitorização***. Assim, após o login por parte do utilizador, esta sincronização é feita automaticamente e todos os dados necessários são passados do servidor para as bases de dados locais da aplicação. Este procedimento é algo muito vantajoso, já que impede que a cada dado que a aplicação necessita seja necessária uma ligação ao servidor. Porém, o utilizador pode sincronizar novamente ambas as bases de dados sempre que quiser, existindo um botão de sincronização para cada uma das bases de dados.

A ***AppMotorista*** requer ligação à internet (pode ser wi-fi, 3G, 4G, etc.) para enviar e receber dados de e para o servidor. Durante um serviço necessita de ligação bluetooth para poder comunicar com o TPA, e necessita ainda de ter o serviço de localização do telemóvel ativado.

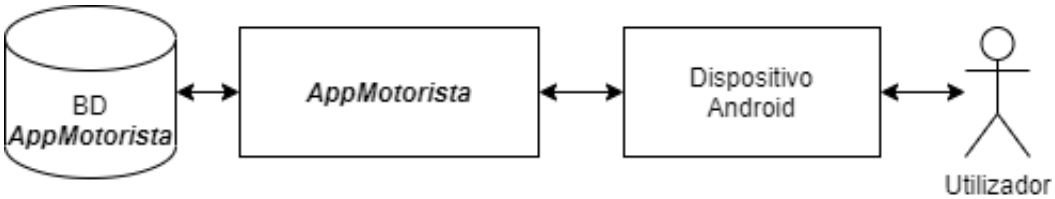

Figura 3.5: Organização do Sistema ***AppMotorista***

3.1.5 AppCliente

A ***AppCliente*** é uma solução de informação ao público para operadores de transporte público, baseada em plataformas mobile (iOS e Android) e Internet (*widget web*/micro site para chamada a partir do site do operador). A solução inclui um módulo de back-office para gestão da informação relativa a paragens e horários;

3.1.6 Gestão Operacional

A ***Gestão Operacional*** é a solução para gestão operacional de transportes em tempo real, tendo em conta diversas dimensões da operação, por exemplo, motoristas, viaturas e manutenção.

3.2 Desafios

Além dos objetivos referidos no 1º capítulo deste relatório, esta dissertação de mestrado contempla mais especificamente os seguintes desafios:

- Correcções e melhorias de funcionalidades na criação de serviços ToD no ***Planeamento***;
- Permitir a exportação de serviços ToD do ***Planeamento*** para a ***Monitorização*** através de uma API;
- Implementação de um ecrã na ***AppMotorista*** onde é possível visualizar os serviços ToD disponíveis, bem como funcionalidades de aceitar, iniciar e concluir serviço;
- Implementar funcionalidade de agregação de pedidos num só serviço;
- Implementação de funcionalidades de serviços ToD na ***AppMotorista*** com ligação ao ***Planeamento***;
- Estudo e implementação de plataformas *low-code* que sirvam de base a uma segunda versão do ***Planeamento***.

Capítulo 4

Trabalho Desenvolvido

4.1 *Planeamento* - Criação de Serviços Ocasionais

A primeira etapa deste projeto, além do tempo de ambientação à tecnologia e à ferramenta ***Planeamento***, consistiu em realizar algumas melhorias na criação de serviços ocasionais (ou ToD). Um serviço ocasional pode ter várias informações:

- Data;
- Cliente (opcional);
- Nº passageiros (opcional);
- Notas (opcional);
- Distância (opcional);
- Portagens (opcional);
- Horas de condução (opcional);
- Cor da viagem (opcional);
- Paragem inicial e hora de início;
- Paragens intermédias e hora de passagem (opcional);
- Paragem final e hora de fim.

Numa primeira fase optou-se pela obrigatoriedade das paragens inicial e final serem paragens reais, enquanto as paragens intermédias poderem ser reais ou temporárias, embora numa fase posterior se preveja que a solução suporte um sistema *door-to-door* desde a origem até ao destino. Paragens reais são paragens que são definidas pelo operador, com um local fixo. Uma paragem temporária é um local definido pelo utilizador e essa paragem apenas é válida para esse serviço, não ficando guardada na base de dados como uma paragem em si, mas apenas como uma localização (com a latitude, longitude e o nome que o utilizador deu à paragem temporária), uma vez que as paragens na base de dados

apenas referem-se a paragens físicas/sinalizadas na via pública, onde no próprio local exista a indicação de que ali existe uma paragem de transportes públicos. Esta opção de paragens temporárias existe para o facto de poderem existir casos em que a paragem mais próxima do passageiro seja muito longe e/ou seja impossível para o passageiro conseguir deslocar-se até ela (por exemplo: pessoas com mobilidade reduzida).

Entre cada paragem podem ser feitos dois cálculos: duração (horas e minutos) e distância (km). Em cada paragem é também possível ver o tempo acumulado e a distância acumulada, ou seja, o tempo/distância acumulados desde a paragem inicial. Estes cálculos são efetuados utilizando uma implementação externa, a OSRM (Open Street Routing Machine).

Este motor calcula a distância e o tempo de viagem entre 2 coordenadas em conjunto com o OpenStreetMap (OSM), e permite ao utilizador realizar pedidos ilimitados. O OSM é uma plataforma de dados abertos que dispõe de diversos tipos de dados referentes a vias, edifícios, entre outros. Dado que as rotas são calculadas utilizando o OSM, normalmente estas terão sempre em conta possíveis desvios de rota devido a obras existentes ou ruas temporariamente fechadas, uma vez que as informações presentes no OSM provêm do contributo de mais de 1 milhão e meio de utilizadores que atualizam e/ou inserem novos dados cartográficos nesta plataforma frequentemente.

Através deste motor é também possível obter uma pré-visualização do percurso do serviço, desde a paragem inicial à paragem final, passando pelas paragens intermédias (caso estejam definidas).

Figura 4.1: Visualização do percurso através do ***Planeamento***

Nas paragens intermédias é também possível alterar a ordem destas.

Antes de serem feitas algumas melhorias na criação de serviços ocasionais, foram realizados testes no sentido de identificar melhorias das funcionalidades atuais, para se proceder posteriormente às respetivas otimizações. As melhorias realizadas foram as se-

guintes:

- O sistema estava desenhado de modo a que o serviço resultasse num percurso circular, ou seja, a paragem final era automaticamente definida, sendo atribuída a paragem que fosse escolhida como paragem inicial, não possibilitando o utilizador de escolher uma paragem final diferente da inicial. Por exemplo, se fosse definido um serviço com 3 paragens em que a paragem número 1 é a inicial, o percurso criado no serviço seria 1-2-3-1.
  - Este cenário foi alterado, passando a ser dada a possibilidade ao utilizador de escolher uma paragem final diferente daquela que foi escolhida como paragem inicial.
- Apesar de na página de edição de serviços ocasionais estarem presentes os campos referentes aos tempos e distâncias parciais e totais, foi necessário implementar a funcionalidade.
  - Os cálculos das distâncias e tempos foram implementados, procedendo-se também à conversão do resultado obtido pela OSRM de metros para quilómetros e de segundos para horas/minutos.
- Foi necessário concretizar o processo para que todos os dados definidos aquando da criação e edição de serviços ocasionais fossem guardados na base de dados do ***Planeamento***.
  - Todas as informações que são definidas no momento da criação/edição de um serviço ocasional já são guardadas na base de dados.
- Após os dados serem guardados na base de dados do ***Planeamento*** (ponto anterior), procedeu-se à criação da estrutura para que estes fossem apresentados na página quando o utilizador pretender editar um serviço ocasional.
  - Todas as informações já são mostradas quando é feito o acesso a um serviço previamente criado e guardado.
- Apesar de na página de edição de serviços ocasionais estar presente um botão para eliminar o respetivo serviço, foi necessário implementar esta funcionalidade.
  - Neste momento, ao clicar no botão de eliminar serviço, o mesmo já é eliminado. O utilizador é levado de volta ao ecrã inicial do sistema.
- Tanto na página onde aparecem listados os serviços ocasionais do operador/unidade/cliente selecionados, como na página de edição de um serviço ocasional, existe um botão que permite exportar o serviço ocasional para a ***MonitorizaçãoAPI***.

Porém, houve a necessidade de se proceder à implementação da exportação através da página de edição de um serviço ocasional.

  - Esta questão foi resolvida e os dados já são exportados para a ***MonitorizaçãoAPI***.

- Foram adicionadas diversas *tooltips* de forma a auxiliar o utilizador a compreender melhor algumas das funcionalidades do sistema. Com esta implementação pretendeu-se cumprir umas das heurísticas da usabilidade criadas por Jakob Nielsen, mais conhecidas por "Heurísticas de Nielsen"[3]. Neste caso, a heurística em questão é a H2-10 "Ajuda e Documentação".
- Na tabela dos serviços ocasionais foram adicionadas duas colunas: uma referente ao nº de paragens intermédias (serviços com apenas paragem inicial e paragem final terão 0 (zero) paragens intermédias); outra referente à distância total do serviço, em quilómetros.
- Foi adicionada uma verificação que averigua se as horas das paragens intermédias e paragem final não são anteriores à hora da paragem inicial. Um serviço ocasional não pode começar num dia e terminar no dia seguinte.
- Foi adicionada a funcionalidade de calcular o WKT de um serviço ocasional. Este WKT é guardado na base de dados quando um serviço ocasional é também ele guardado. WKT significa "Well-Known Text"e trata-se de uma linguagem de marcação para representar objetos geométricos vetoriais. O WKT contém 7 tipos diferentes de dados, sendo que neste caso o utilizado é o tipo LineString. Este é um exemplo da informação guardada na base de dados, contendo 5 coordenadas:

  GEOMETRYCOLLECTION(LINESTRING(-9.419702 38.701681,-9.419795

  38.701694,-9.419675 38.701776,-9.419649 38.701805,-9.419624 38.701833))

  A construção é feita da seguinte forma:

  $$GEOMETRYCOLLECTION(LINESTRING(lon_1\ lat_1, ..., lon_n\ lat_n))$$

  Onde n é o nº total de coordenadas que compõem o trajeto do percurso.
- Foi adicionada a opção de fazer o download de um ficheiro KML referente ao percurso do serviço ocasional. Este ficheiro KML pode ser aberto em diversos softwares, como por exemplo o Google Earth, onde o operador poderá visualizar o percurso de diferentes formas e obter mais informações. Esta funcionaliade foi implementada através da construção faseada de um ficheiro XML, utilizando o namespace System.XML. Neste ficheiro KML são incluídas as paragens inicial,

[3] https://medium.com/aela/10-heur%C3%ADsticas-de-nielsen-dicas-para-melhorar-a-usabilidade-de-sua-interface-35ef86a7fb41

intermédias (a existirem) e final, bem como o trajeto do serviço. A paragem inicial tem um ícone verde, as paragens intermédias reais têm um ícone amarelo, as paragens intermédias temporárias têm um ícone azul e a paragem final tem um ícone vermelho. Cada ícone tem ainda uma legenda que inclui o nome da paragem e o identificador único desta. No caso de se tratar de uma paragem temporária, tem também esta indicação. Na figura 4.2 é possível ver o percurso através da visualização do ***Planeamento***, e na figura 4.3 o mesmo percurso no Google Earth através do ficheiro KML que foi exportado.

Figura 4.2: Visualização do percurso através do ***Planeamento***

Figura 4.3: Visualização do percurso através do Google Earth

- No ecrã de editar um serviço ocasional, para facilitar o trabalho ao utilizador, foi adicionado ao lado de cada *textbox* da hora de passagem pela paragem (intermédias e final) um botão de copiar que, quando o utilizador clica, é copiado para este *textbox* o valor da hora de passagem sugerida pela OSRM. Na figura 4.4 é possível ver este botão em cada uma das linhas. Neste caso são apresentadas 4 paragens intermédias, sendo que apenas nas duas primeiras o botão de copiar a hora sugerida foi clicado.

Sugestão de hora	Hora
14:02	14:02
14:05	14:05
14:06	14:15
14:07	14:20

Figura 4.4: Botão copiar hora sugerida.

4.2 Exportação de Serviços Ocasionais

Antes de serem feitas alterações e/ou implementadas novas opções na exportação de serviços ocasionais, foi necessário estudar o percurso desde que um serviço ocasional é criado no ***Planeamento*** até que este chega à ***AppMotorista***.

Após um serviço ocasional ser criado no ***Planeamento***, existe a opção deste ser exportado para a ***Monitorização***. Após ser acionada esta opção, é chamada a ***MonitorizaçãoAPI***, que irá escrever diretamente na base de dados da ***Monitorização*** as informações deste serviço ocasional, utilizando a ferramenta *Entity Framework*.

Esta é uma API que está assente numa arquitetura do tipo MVC, e faz a ligação entre o ***Planeamento*** e a ***Monitorização***. A ***MonitorizaçãoAPI*** contém 3 entidades às quais é possível efetuar pedidos: OnDemand, Stop e TransportationService. Na tabela 4.1 são apresentados os possíveis pedidos a cada uma destas entidades.

Serviço	**GET**	**PUT**	**POST**	**DELETE**
OnDemand	/api/ondemand /api/ondemand/{id}		/api/ondemand	/api/ondemand{id}
Stop	/api/stops /api/stops/{id}	/api/stops/{id}	/api/stops	/api/stops/{id}
TransportationService	/api/services /api/services/{id} /api/services/available		/api/services/{id}/action	

Tabela 4.1: Pedidos por cada entidade da API

Após o estudo da ligação entre o ***Planeamento*** e a ***MonitorizaçãoAPI*** foi verificado que os serviços ocasionais não ficavam todos na ***Monitorização***, uma vez que existia uma falha na verificação de serviços duplicados. Esta falha fazia com que apenas o último serviço adicionado ficasse guardado, visto que o serviço a ser adicionado substituía sempre o último serviço que tinha sido guardado anteriormente, fosse duplicado ou não. Depois de corrigida esta falha na verificação de serviços duplicados, passaram a ser guardados todos os serviços que não eram duplicados. Foi necessário proceder a duas novas implementações

- Inicialmente, caso um serviço ocasional tivesse paragens intermédias, este não era exportado, ou seja, a exportação apenas funcionava caso o serviço a ser exportado tivesse somente paragem inicial e paragem final. Neste momento a exportação já funciona para ambos os casos, ou seja, tanto os serviços que possuem paragens intermédias como os que não possuem, são exportados corretamente.

- Depois de realizada a melhoria anterior, verificou-se que não era possível adicionar serviços em que as paragens intermédias fossem paragens reais (apenas paragens temporárias). Esta questão foi também resolvida.

Daqui em diante, isto é, a partir do sistema ***Monitorização***, foi necessário construir toda a lógica no que aos serviços ocasionais diz respeito.

O primeiro passo foi permitir que a API ***AppMotoristaAPI*** conseguisse obter os serviços ocasionais que estão na base de dados da ***Monitorização***. Esta é a API que permite a comunicação entre a ***Monitorização*** e a ***AppMotorista***. Para isto ser possível, foi necessário criar um novo SP na base de dados da ***Monitorização*** que fosse buscar à base de dados da ***Monitorização*** os serviços ocasionais planeados, ou seja, este SP vai obter todos os serviços que são do tipo ocasional (que neste caso, na base de dados, corresponde ao tipo nº4).

Foi também necessário proceder à criação de um novo *webservice* através do qual o sistema ***AppMotorista*** irá posteriormente obter os serviços ocasionais a fim de colocar a informação sobre estes serviços na base de dados da ***AppMotorista***.

Depois de configurada a obtenção dos serviços ocasionais por parte da ***AppMotoristaAPI***, foi necessário implementar a lógica na aplicação ***AppMotorista*** que possibilita ao utilizador a visualização das informações destes serviços ocasionais, bem como a sua inicialização e respetivo término.

Em primeiro lugar, foi necessário criar a lógica que permite à aplicação obter os serviços ocasionais existentes. Esta obtenção é realizada através da ***AppMotoristaAPI***, utilizando o *webservice* criado anteriormente para este efeito. Depois destes serviços serem obtidos, são colocados na base de dados local da aplicação móvel.

Depois, foi necessário criar a lógica para serem apresentados os serviços ocasionais. Esta etapa passou por adicionar uma nova opção ao spinner que aparece quando se seleciona a opção de iniciar serviço. Esta nova opção tem o título ”ToD”.

De seguida, foi necessária a criação de 2 novos ficheiros:

1. O primeiro - *TODServiceBuilder* - contém o código necessário para construir um serviço ocasional. Esta construção é feita após o utilizador selecionar o tipo de serviço que quer utilizar (através do spinner referido anteriormente), até ao momento em que o serviço é iniciado. Neste processo, no caso dos serviços ocasionais, o utilizador apenas tem de escolher o serviço que irá realizar, não existindo parâmetros como a linha, o sentido ou o horário, como acontece noutras situações. Após o utilizador selecionar o serviço que irá realizar, as informações deste serviço selecionado serão enviadas para *TODServiceBuilder* e o serviço será construído. Nesta fase o utilizador tem a possibilidade de adicionar algumas notas sobre o serviço (no caso dos serviços regulares existem diversos horários para uma linha/-sentido).

2. O segundo - *TODServiceBuilderEvent* - contém o código para obter ou definir parâmetros sobre o serviço, tais como obter/definir a linha, a circulação, o serviço ou verificar/definir se o serviço está completo ou não.

Após ser finalizada a lógica do lado da aplicação móvel, foram efetuados diversos testes. Durante estes testes, foi verificado que era necessário acrescentar do lado do ***Planeamento*** uma indicação à base de dados da ***Bilhética*** de que foram adicionados novos serviços (isto para o caso dos serviços serem realizados no mesmo dia em que foram criados). Isto acontece pois cada serviço ocasional criado corresponde a uma nova linha, e assim é sempre necessário acrescentar novos dados de bilhética à base de dados, dados sem os quais não é possível iniciar um serviço na aplicação móvel. Assim, sempre que ocorre uma exportação de um serviço ocasional, o atributo "*LastUpdate*"referente a 4 tabelas da base de dados da ***Bilhética*** é atualizado para o *timestamp* em que a exportação foi efetuada, tabelas essas que se referem a:

1. Linha do serviço;
2. Paragens do serviço;
3. Áreas de bilhética do serviço;
4. Espinha do serviço.

Finalizados todos os testes das implementações feitas, foi assim concluído o processo desde a criação de um serviço ocasional no sistema ***Planeamento*** até ao início e respetivo término dentro da aplicação móvel ***AppMotorista***.

4.3 Execução de Serviços Ocasionais

Terminado o processo desde a criação de um serviço ocasional no ***Planeamento*** até ao seu início e respetivo término na ***AppMotorista*** que foi descrito na secção anterior, foi necessário olhar para a lógica do envio de informações no sentido inverso a partir do momento em que um motorista inicia um serviço, ou seja, desde a ***AppMotorista*** até ao ***Planeamento***, passando pela ***Monitorização***.

A lógica desde a ***AppMotorista*** até à ***Monitorização*** já estava implementada, e , resumidamente, funciona da seguinte maneira: quando na ***AppMotorista*** é iniciado ou findado um serviço, a ***AppMotorista*** comunica essa ação à ***AppMotoristaAPI***, enviando o identificador único da circulação, o *timestamp* e o código do motorista.

Dependendo da ação (iniciar circulação ou terminar circulação), é executado um método na ***AppMotoristaAPI*** que irá chamar um SP da base de dados da ***Monitorização***.

Este SP, após algumas verificações e caso estas não falhem, irá inserir os dados sobre o início ou fim da circulação na base de dados da ***Monitorização***. Este SP vai retornar um número inteiro. Se as verificações forem bem sucedidas, vai retornar o identificador da circulação; caso contrário, retorna -1. De seguida é explicado o restante processo que foi implementado com o objetivo de fazer chegar estes dados até ao ***Planeamento***.

Pegando no valor retornado pelo SP anteriormente referido, caso este seja maior que zero (ou seja, terá sido retornado o identificador da circulação), vai ser verificado o tipo deste serviço. Se o tipo for ”OnDemand”(ou seja, um serviço ocasional), vai ser executado um novo método que foi implementado: *AcceptOrTerminateDriverServiceInPlanning*. Este método recebe diversos dados: identificador único do cliente (*IDCustomer*), do equipamento (*IDVehicle*), do condutor (*IDDriver*), da circulação (*IDCirculation*), a hora de início da circulação (em UTC) (*StartTimeUTC*), a hora de fim da circulação (em UTC) (*EndTimeUTC*) e a ação (*Action*). A hora de início da circulação só estará definida caso seja um pedido para iniciar serviço, caso contrário este valor será nulo (a mesma coisa para o caso da hora de fim, caso o pedido seja para terminar serviço). A variável ação pode tomar dois valores: ”Accept”(caso seja para iniciar serviço) ou ”Terminate”(caso seja para terminar serviço).

Depois do método receber os dados acima referidos, vão ser feitas ligações à base de dados para se obterem informações sobre o operador, o equipamento que está instalado na viatura e ainda sobre o condutor. Se todos estes dados existirem (não forem nulos) vai ser criado um objeto de uma nova classe que foi implementada, cuja constituição está patente na figura 4.5.

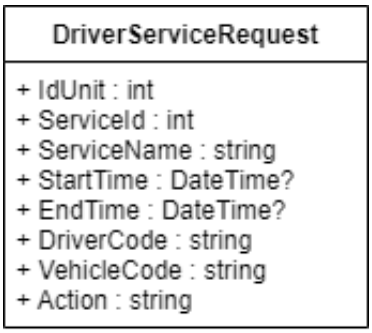

Figura 4.5: Diagrama da classe DriverServiceRequest

Na criação do objeto, dependendo do valor da *Action*, o *StartTime* e o *EndTime* podem tomar diferentes valores.

- *Action* = ”Accept”:
  - *StartTime* ficará com o valor recebido pelo método que está na variável *StartTimeUTC*.
  - *EndTime* ficará com o valor da hora de fim planeada do serviço.
- *Action* = ”Terminate”:
  - *StartTime* ficará com o valor de início de serviço atribuído anteriormente, quando o serviço foi iniciado (uma vez que para terminar um serviço este já terá sido iniciado e, consequentemente, já estará atribuída uma hora de início).
  - *EndTime* ficará com o valor recebido pelo método que está na variável *EndTimeUTC*.

Criado o objeto DriverServiceRequest, este será enviado na chamada à API do ***Planeamento*** que é feita de seguida, através de um pedido do tipo POST.

Já na API do ***Planeamento***, utilizando os dados incluídos no objeto DriverServiceRequest que foi enviado, serão feitas algumas operações com ligação à base de dados do ***Planeamento***.

Na descrição seguinte são mencionados 2 objetos: *TaskSequence* - objeto que atribui um serviço à chapa do condutor, bem como a viatura a ser usada; *TaskTimeTableColumn* - objeto responsável por ligar o *TaskSequence* a uma circulação de uma linha.

Dependendo do valor da *Action*, teremos 2 casos diferentes:

- *Action* = ”Accept”:
  - Vai ser verificado se já existe na base de dados do ***Planeamento*** um *TaskTimeTableColumn* referente ao serviço em questão. Caso não exista, é criado um *TaskSequence* e também um *TaskTimeTableColumn* utilizando os dados do objeto DriverServiceRequest. Caso já exista um *TaskTimeTableColumn* é retornado o aviso de que este serviço já foi aceite anteriormente e a execução termina.
- *Action* = ”Terminate”:
  - Vai ser verificado se já existe um *TaskTimeTableColumn* referente ao serviço em questão, algo que é necessário para a execução deste processo pois para um serviço ser terminado já terá de ter sido iniciado, o que implica a existência de um *TaskTimeTableColumn*. Assim, caso não exista, é retornado erro a dizer que não existe. Para o caso de existir um *TaskTimeTableColumn*, é verificado se o identificador único do condutor presente no DriverServiceRequest é igual ao que está atribuído no *TaskTimeTableColumn*. Para o caso de não serem iguais, será retornado o erro a dizer que este condutor (do DriverServiceRequest) não pode terminar este serviço. Caso os identificadores coincidam, o campo referente à hora de fim do serviço do *TaskTimeTableColumn* é atualizado, ficando com o valor da variável *EndTime* que está no objeto DriverServiceRequest que foi recebido pela API.

Inseridos e atualizados na base de dados do ***Planeamento*** todos os dados referentes ao início e fim de um serviço, podemos visualizar graficamente essas informações através de um ecrã já existente denominado *Graficagem*.

4.4 Agregação de Pedidos

Neste capítulo é abordado o tema da Agregação de Pedidos. Agregação de pedidos é uma funcionalidade que consiste em juntar 2 ou mais serviços ocasionais existentes num só. Esta optimização fornece diversos benefícios, não só para o operador, como para os passageiros e para a sociedade em geral.

- Traz benefícios para o operador pois acaba por haver uma redução de custos, uma vez que são percorridos menos quilómetros, são necessárias menos viaturas e são necessários menos meios humanos para que a operação seja realizada. Isto ajuda a que as empresas consigam ter uma melhor sustentabilidade económica e financeira, evitando até certo ponto o seu encerramento.

- Traz benefícios para os passageiros porque, uma vez que há uma redução de custos por parte da empresa, muito provavelmente o custo das viagens para os passageiros também será menor.

- E traz benefícios para a sociedade em geral pois haverão menos viaturas a circular, logo reduz o risco de existência de trânsito e por sua vez menos emissões poluentes serão realizadas, ajudando assim também o meio ambiente.

Nos parágrafos seguintes serão colocados números à frente de algumas das funcionalidades implementadas, que posteriormente irão ser representadas na figura 4.6.

Foi criada uma nova página na interface web do sistema *textbfPlaneamento* onde são dispostos numa tabela todos os serviços ocasionais daquele operador/unidade para aquele dia/hora, de todos os clientes. Por defeito, quando a página é aberta, são mostrados todos os serviços para o próprio dia. Porém, o utilizador tem a opção de filtrar os resultados por data e por hora de início e/ou fim (1).

Foi adicionado um botão "*Voltar*"que permite ao utilizador voltar ao ecrã anterior (2), não obrigando o utilizador a ir para a página inicial caso se tenha enganado a clicar no botão ou caso queira regressar à página anterior.

Nesta tabela existe ainda a possibilidade de filtrar os serviços agregados, isto é, de apenas serem visualizados aqueles que não são serviços agregados (3).

Em cada linha da tabela, que corresponde a um único serviço ocasional, existe a indicação de que se o serviço em si já é um serviço agregado ou não (4), e existe ainda uma checkbox onde o utilizador poderá selecionar os serviços ocasionais que pretende agregar (5). Depois de selecionados os serviços ocasionais, o utilizador terá de carregar num botão que tem a denominação "Agregar" (6). Depois terá de aguardar uns segundos até ter a confirmação (ou não) de que o pedido de agregação dos serviços ocasionais

selecionados foi bem sucedido. Depois deste botão ”Agregar”foi adicionada uma opção em que o utilizador vai decidir se deixa o algoritmo ter o ”poder”de decidir se o serviço agregado será criado ou não (7). Esta decisão tem por base uma verificação assente numa função matemática que será explicada mais à frente. Ao lado desta opção foi adicionado um ícone azul de ajuda que, o utilizador ao colocar o rato sobre ele, irá exibir uma *tooltip* onde é explicado o que é esta opção de deixar o algoritmo decidir (8). Por fim, foi adicionado no canto inferior direito um botão ”Ajuda”que, ao ser clicado, irá apresentar um *pop-up* com uma breve descrição de como a agregação de serviços funciona (9).

Na figura 4.6 é possível visualizar as funcionalidades acima descritas, fazendo corresponder os números atribuídos a cada uma delas.

Figura 4.6: Página de Agregação de Pedidos no sistema ***Planeamento***

Quando é enviado um pedido de agregação dos serviços serão feitos alguns cálculos de modo a tentar optimizar o percurso do serviço agregado. Este serviço agregado irá conter todas as paragens de todos os serviços ocasionais selecionados. Porém, se uma ou mais paragens estiverem presentes em mais que 1 serviço, o serviço agregado apenas irá passar 1 vez por esta paragem.

Existem 2 principais critérios para a criação do percurso de serviço ocasional agregado:

- O serviço agregado percorre todas as paragens presentes nos serviços ocasionais selecionados anteriormente;
- O serviço agregado tem de respeitar a ordem das paragens de cada serviço, isto é, se um serviço fizer o percurso A →B, o serviço agregado não poderá percorrer a paragem B antes da paragem A;

Para a agregação de serviços foi escolhido o algoritmo *nearest neighbor*. Este algoritmo é simples e intuitivo, e resume-se a encontrar qual o próximo ponto a visitar, que será aquele que está mais próximo. Este algoritmo vai construindo a solução à medida que

vai sendo executado. O *nearest neighbor* não explora a totalidade de soluções possíveis, portanto a solução encontrada pode não ser a solução óptima. Porém, a sua execução é mais rápida do que se fosse usado um algoritmo onde fosse explorada a totalidade das soluções possíveis a fim de encontrar a solução óptima. A solução encontrada apesar de poder não ser óptima, será ainda assim uma boa solução e esta será encontrada num espaço de tempo razoável em comparação com algoritmos de pesquisa exaustiva que verificam todas as possíveis soluções até encontrarem a solução óptima, e a sua implementação acaba por ser mais simples do que nestes.

O algoritmo *nearest neighbor* pode ser resumido nos seguintes passos. Para esta descrição os vértices serão paragens de autocarro:

1. Iniciar todas as paragens como não visitadas;
2. selecionar uma paragem aleatória e defini-la como paragem atual ***A***. Marcar ***A*** como visitada.
3. Calcular a distância entre ***A*** e cada paragem ***V*** que ainda não tenha sido visitada.
4. Ligar ***A*** à paragem ***V*** cuja distância é a mais pequena entre as calculadas no ponto anterior.
5. Definir ***V*** como paragem atual ***A***. Marcar ***V*** como visitada.
6. Caso todas as paragens estejam marcadas como visitadas, o processo termina. Caso contrário, irá voltar para a etapa nº 3.

Este poderia ser um processo a adoptar caso esta funcionalidade se tratasse de entrega de encomendas, por exemplo, onde a ordem de entrega das encomendas não interessa. Porém, existem diversas nuances que têm de ser incluídas no algoritmo, tornando-o um pouco mais complexo.

O primeiro detalhe é que esta funcionalidade trata-se de viagens com passageiros, onde a ordem entre as paragens dos serviços originais (a agregar) tem de ser respeitada. Por exemplo, se num serviço o passageiro quer ir da paragem A à paragem B, no serviço agregado a paragem B não pode ser percorrida antes da paragem A.

Outra nuance é que, como já foi referido anteriormente, neste momento as paragens inicial e final têm de ser paragens reais, isto é, paragens que são definidas pelo operador. No caso de existirem paragens temporárias nos serviços que irão ser agregados, estas não poderão ser nem a paragem inicial, nem a paragem final.

O conceito ”paragem-mãe”utilizado na descrição seguinte refere-se à paragem anterior de uma qualquer paragem de um serviço ocasional. Se um serviço ocasional fizer o percurso A →B →C, a paragem-mãe de B é A, a paragem-mãe de C é B, e A não tem paragem-mãe.

Tendo as duas nuances anteriormente descritas em consideração, o processo implementado é resumidamente o seguinte:

1. Iniciar todas as paragens como não visitadas. Todas as paragens têm a indicação se são paragens reais ou não, e qual é a paragem-mãe referente ao serviço original (para o caso de se tratar de uma paragem inicial, este campo é colocado a *null*).

2. A primeira paragem escolhida ***A*** corresponde à paragem inicial do serviço que tem o seu início mais cedo. Marcar ***A*** como paragem atual e como visitada.

3. Se restarem 3 paragens por visitar e alguma dessas paragens ***V*** for uma paragem temporária:

   (a) Se a paragem-mãe de ***V*** ainda não tiver sido visitada:

      i. Ligar ***A*** à paragem-mãe de ***V***. Marcar a paragem-mãe de ***V*** como visitada;
      ii. Ligar a paragem-mãe de ***V*** a ***V***. Marcar ***V*** como visitada;
      iii. Ligar a única paragem que falta a ***V***, e marcar como visitada;
      iv. Terminar o processo.

4. Se restarem 2 paragens por visitar e algumas dessas paragens ***V*** for uma paragem temporária (o processo só entra neste ponto caso não tenha entrado no ponto 3):

   (a) Ligar ***A*** a ***V***. Marcar ***V*** como visitada;
   (b) Ligar a única paragem que ainda não está visitada a ***V***, e marcar como visitada;
   (c) Terminar o processo.

5. Calcular a duração do trajeto entre ***A*** e cada paragem ***V*** que (o processo só entra neste ponto se não tiver entrado no ponto 3 ou 4):

   (a) Ainda não tenha sido visitada, e:

      i. Não tenha paragem-mãe, ou;
      ii. A paragem-mãe esteja marcada como visitada.

6. Ligar ***A*** à paragem ***V*** cuja duração é a menor entre ***A*** e todas as paragens ainda por visitar.

7. Marcar ***V*** como visitada. Definir ***V*** como paragem atual ***A***.

8. Se todas as paragens estiverem marcadas como visitadas, terminar o processo. Caso contrário, voltar à etapa 3.

O cálculo das durações dos trajetos entre paragens é efetuado através da OSRM.

Inicialmente o algoritmo tinha sido implementado tendo em conta a distância entre paragens. Porém, após alguma reflexão, verificou-se que fazia mais sentido ter em conta a duração e portanto decidiu-se alterar o algoritmo para construir o serviço agregado através dos trajetos que demorem menos tempo, ao invés da distância.

Após a rota optimizada estar construída, será feita uma verificação que irá comparar o resultado obtido com os serviços originais a serem agregados. Para esta verificação será usada uma função matemática, onde a variável x refere-se à duração de um dado percurso original, e o resultado será a percentagem máxima da duração que o novo serviço poderá ter em relação ao serviço original em questão. Por exemplo, se no serviço original a duração entre A e B for de 10 minutos e o resultado da função for 122, o serviço agregado só será criado se a duração entre A e B no novo percurso for igual ou inferior a 122% de 10 minutos (que são 22,2 minutos = 22 minutos e 12 segundos).

A função matemática utilizada traduz-se na seguinte expressão:

$$-100\log(\tfrac{x}{2}+1)+200$$

Na figura 4.7 é possível ver a representação desta função num plano cartesiano ortogonal. É possível verificar que à medida que a duração aumenta, a percentagem limite diminui.

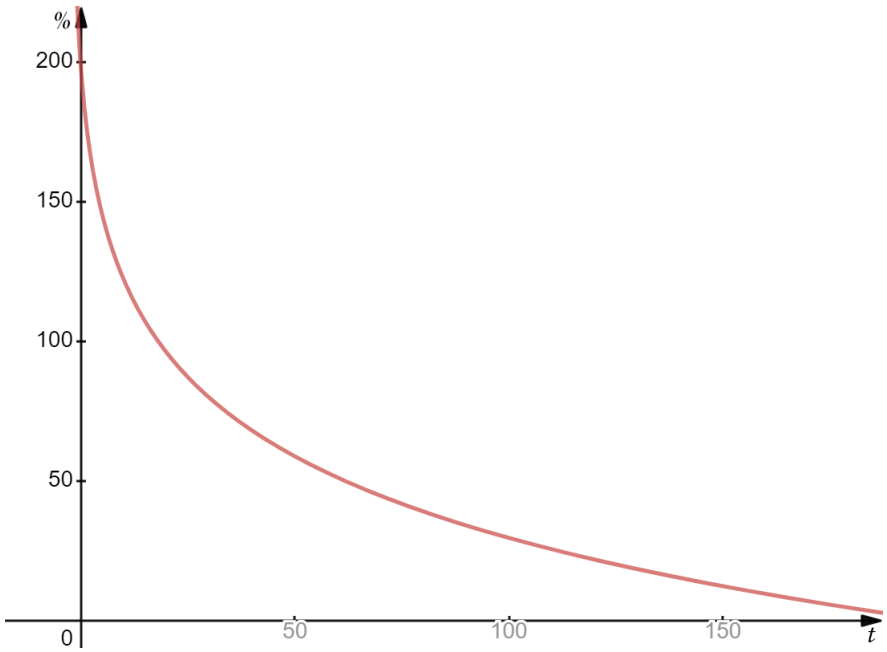

Figura 4.7: Gráfico da Função

Esta função foi criada com o intuito de reduzir a duração máxima entre duas paragens em função da duração original, à medida que a duração aumenta.

Na tabela 4.2 é possível verificar o resultado da função relativamente a alguns valores de x.

Duração Original (min.)	Resultado (%)	Duração Limite (min.)
1	182.39	2.82
2	169.90	5.40
5	145.59	12.28
10	122.19	22.22
30	79.59	53.88
60	50.86	90.52

Tabela 4.2: Resultados da função em função da duração original

Apenas depois desta verificação ser concluída é que o serviço agregado irá ser criado e guardado na base de dados do sistema. Este serviço ficará referenciado como um serviço agregado, para facilitar a gestão de serviços ao utilizador.

Depois deste processo é enviada uma mensagem de retorno ao utilizador a informar se o serviço foi agregado ou não, como é possível verificar nas figuras 4.8 e 4.9. Esta informação é apresentada no canto inferior esquerdo da página. Esta funcionalidade integra um dos princípios de Norman [4], neste caso aquele que se refere ao *feedback*, onde é mencionado que o utilizador deve receber alguma resposta face a alguma ação que tenha realizado.

Figura 4.8: Mensagem de serviço agregado com sucesso.

Figura 4.9: Mensagem de falha ao agregar o serviço.

Depois do serviço estar criado, o cliente tem a total liberdade para alterar a ordem das paragens, adicionar e/ou remover paragens ao serviço através da página de edição de serviços ocasionais, uma vez que em termos de usabilidade os serviços agregados acabam por ser iguais aos serviços não agregados. Para auxiliar o utilizador a saber quais os clientes e respetivas paragens que um serviço agregado inclui, na página de edição de serviços serão exibidas essas informações no campo das ”Notas”.

[4] https://medium.com/aela/design-de-intera%C3%A7%C3%A3o-os-6-princ%C3%ADpios-fundamentais-d2cb1e585cad

4.5 Funções de Apoio à Operação

O trabalho realizado no âmbito deste capítulo reúne um conjunto de alterações e novas funcionalidades implementadas nos sistemas ***Monitorização*** e ***AppMotorista***, com vista a fornecer mais informação aos utilizadores destes sistemas, melhorando as tomadas de decisão por parte destes.

4.5.1 *Monitorização*

- Foram alteradas algumas *tags* e títulos das páginas que estavam sob o formato *hard-code*:
  - "_Alerts"→"Alertas";
  - "_Line"→"Linha";
  - "_Way"→"Sentido";
  - "_Fleet"→"Viatura";
  - "_Driver"→"Motorista";
  - "_Cellphone"→"Nº Telemóvel";
  - "_AlertsSource"→"Fonte do Alerta";
  - "_AlertsDescription"→"Descrição do Alerta";
  - "_Contact"→"Contactar";
  - "_Progression"→"Progresso";
  - "_Bus"→"Autocarro".
- O ecrã de comparação de horários foi melhorado:
  - Foi adicionado um novo esquema de cores, mais "*friendly*". O significado de cada cor está visível na legenda que também foi adicionada.
- No menu *Gerir Serviços* foi adicionada a opção de ordenar os resultados por qualquer um dos atributos que está disponível na tabela que é apresentada na página. Na figura 4.10 é possível ver essa opção.

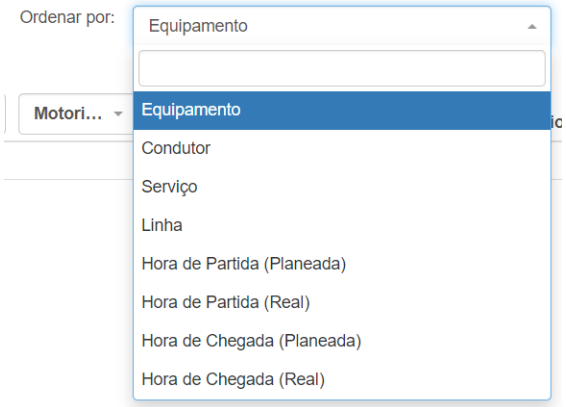

Figura 4.10: Ordenar serviços

- Foi adicionada uma tabela onde se podem verificar quais os serviços que estavam planeados para aquele dia e que não foram realizados (até à hora da consulta dos dados). Esta verificação é realizada através de um pedido à base de dados, por meio de um SP, onde, sumariamente, são obtidos os serviços planeados para o dia em questão e que, até à hora deste mesmo pedido, a hora de início desse serviço bem como o identificador único do motorista associado a esse serviço estejam a *null* (ou seja, o serviço ainda não foi iniciado nem há um motorista atribuído). Na tabela é possível verificar o código da linha, o nome da linha e ainda o horário cujo serviço não foi realizado. Nesta página é ainda possível escolher a data para a qual se pretendem exibir os serviços não realizados. Esta nova tabela está representada na figura 4.11.

Código Linha	Nome da Linha	Horário
1234	Serviço ABC	23/10/2020 06:35:00
1234	Serviço ABC	23/10/2020 12:30:00
1234	Serviço ABC	23/10/2020 15:30:00
1234	Serviço ABC	23/10/2020 16:15:00
1234	Serviço ABC	23/10/2020 17:30:00
5678	Serviço DEF	23/10/2020 08:00:00
5678	Serviço DEF	23/10/2020 08:05:00
5678	Serviço DEF	23/10/2020 08:15:00
5678	Serviço DEF	23/10/2020 08:20:00
5678	Serviço DEF	23/10/2020 08:30:00

Figura 4.11: Ecrã de serviços não realizados.

- No menu *Mapa Dashboard*, que é um menu onde é possível ver num mapa as últimas posições enviadas pelas viaturas e obter alguns detalhes destas, os ícones das viaturas apareciam a verde se a respetiva viatura tinha estado ligada nas últimas

24 horas. Este critério foi alterado, e de momento uma viatura aparece com um ícone verde apenas se estiver a efetuar uma circulação planeada. As restantes viaturas aparecem com um ícone vermelho na última posição que foi recebida pelo sistema.

- No menu *Progresso* foram adicionadas e/ou melhoradas as seguintes funcionalidades:

  - Ao clicar num ícone do autocarro que está numa espinha, foi adicionada a funcionalidade de aparecer um *pop-up* onde são exibidas algumas informações sobre o serviço, autocarro e condutor. Quando se clica no ícone, são mostradas algumas informações. Porém, o utilizador pode expandir estas informações e serão exibidos mais alguns dados. O *pop-up* está representado esquematicamente nas figuras 4.12 e 4.13, sendo que o primeiro refere-se ao *pop-up* que aparece quando o utilizador clica em cima do ícone do autocarro, e o segundo são as informações disponibilizadas após o utilizador expandir as 3 secções do menu.
  - Neste menu existe a possibilidade de filtrar os serviços/viaturas em relação à sua conformidade com o horário planeado, ou seja, podemos filtrar as viaturas que estão adiantadas, dentro do horário, atrasadas e ainda as que estão a fazer serviços não planeados (denominados ”wild”). Acontece que as informações desta página são atualizadas a cada 5 segundos, e depois de ocorrer uma atualização o filtro deixava de estar ativo (ou seja, eram mostrados todos os serviços novamente). A alteração feita aqui foi permitir que o filtro continuasse ativo mesmo depois dos dados serem atualizados;
  - Por baixo de cada paragem pela qual a viatura já passou é exibido o número de minutos de adiantamento ou atraso. No caso da viatura passar à hora planeada, não é exibido. As cores com que eram apresentados estes minutos não correspondiam às cores que são usadas nos ícones das viaturas: os minutos apareciam a vermelho no caso da viatura estar atrasada, e a verde no caso de estar adiantada, sendo que nos ícones a cor vermelha é atribuída ao caso da viatura estar adiantada e para o caso de estar atrasada é usada a cor amarela. Assim, as cores foram corrigidas: minutos de atraso passaram a ser representados a amarelo ”torrado”(RGB (255,217,0)); minutos de adiantamento passaram a ser representados a vermelho (RGB (255,0,0)); foi adicionada uma nova cor - o verde (RGB (0,255,0)) - que é apresentado sempre que a viatura está dentro do limite de avanço e/ou atraso permitido pela empresa.

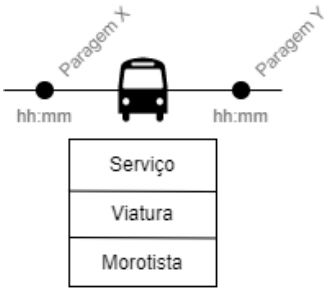

Figura 4.12: *pop-up*.

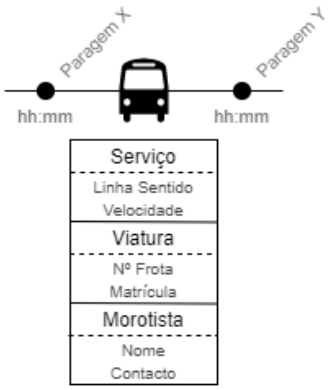

Figura 4.13: *pop-up* expandido.

Limite de Velocidade, Desvio de Percurso e Tempo Médio de Paragem

Numa página já existente da ***Monitorização*** foram adicionados 3 novos tipos de informação que podem ser visualizados num mapa. Estas 3 novas informações são:

- Desvios de percurso;
- Limite de velocidade excedido;
- Tempo de paragem por paragem.

Foram criadas 3 tabelas na base de dados para albergar a informação sobre os limites de velocidade excedidos, desvios de rota e os tempos médios de paragem por paragem, cujas colunas e respetivos tipos de dados são mostrados nas figuras 4.14, 4.15 e 4.16.

SpeedExceeded	
PK	- ID : int
	- LineServiceID : int - EquipmentID : uniqueidentifier - RegistrationLatitude : float - RegistrationLongitude : float - RegistrationTimestamp : datetime - Speed : float - DriverName : nvarchar(100)

Figura 4.14: Tabela SpeedExceeded

RouteDeviations	
PK	- ID : int
	- LineServiceID : int - RegistrationLatitude : float - RegistrationLongitude : float - Distance : float - WKTLatitude : float - WKTLongitude : float - RegistrationTimestamp : datetime - EquipmentID : uniqueidentifier - DriverName : nvarchar(100)

Figura 4.15: Tabela RouteDeviations

MeanStopTime	
PK	- ID : int
	- LineServiceID : int - EquipmentID : uniqueidentifier - StopID : int - StopLatitude : float - StopLongitude : float - RegistrationLatitude : float - RegistrationLongitude : float - DistanceStopLocation : float - RegistrationTimestamp : datetime - ArrivalDateTime : datetime

Figura 4.16: Tabela MeanStopTime

A informação que é inserida nestas 3 tabelas provém de um pré-processamento que é efetuado na ***MonitorizaçãoAPI***, sempre que um autocarro envia um conjunto de posições (sensivelmente de 30 em 30 segundos, e cada conjunto contém informações recolhidas sensivelmente de 5 em 5 segundos). Estas posições que são enviadas pelas viaturas contêm as seguintes informações:

- Altitude;
- Identificador único do equipamento;
- Endereço IP do equipamento;
- Latitude;
- Longitude;
- Velocidade;
- Data e hora.

A cada posição do autocarro que é enviada, é verificada a velocidade do mesmo. Se a velocidade exceder um certo limite definido (por exemplo, 100km/h), esta informação é colocada na tabela SpeedExceeded.

Para os cálculos dos desvios de rota e dos tempos de paragem por paragem, são feitos 2 pedidos à base de dados da ***Monitorização*** através de um SP.

Para este pedido são enviadas as informações que o autocarro enviou: altitude, identificador único do equipamento, endereço IP do equipamento, latitude, longitude, velocidade e data/hora.

Dentro do SP são feitos 2 pedidos à base de dados:

1. O primeiro pedido vai retornar os dados para que sejam efetuados os cálculos do tempo de paragem por paragem. Para isto é feito um pedido à base de dados que, tendo em conta o identificador do equipamento e a hora de registo da posição da viatura, irá retornar 2 linhas da base de dados: a última paragem pela qual a viatura passou, e a próxima paragem pela qual a viatura irá passar. A forma como se descobre qual é o serviço que a viatura está a fazer resume-se à obtenção do serviço cujo identificador da viatura é igual àquele que foi enviado, e em que o *timestamp* enviado pela viatura está entre a hora real de início e a hora de fim do serviço (que neste caso será *null*, uma vez que o serviço ainda não terminou). Para se obter a paragem anterior e a paragem seguinte é verificada qual é a última paragem da sequência de todas as paragens deste serviço em que a flag ”IsReached”é igual a 1. Assim, é retornada a linha correspondente a essa paragem, bem como a linha seguinte. Esta flag ”IsReached”indica se a viatura já passou pela paragem em questão, e só pode ter 2 valores: 1 caso a viatura já tenha passado, 0 caso ainda não tenha.

2. Tendo em conta o ID do equipamento e a data/hora do registo, são obtidas as informações sobre o serviço que este está a realizar. Aqui a informação fundamental a obter é o WKT do serviço em questão, pois é esse o dado que será necessário para se averiguar se existiram ou não desvios de rota.

O processamento sobre os desvios de rota é efetuado da seguinte forma:

A cada posição do autocarro que é enviada, vai ser calculada a distância entre as coordenadas do autocarro e cada ponto do LineString. Após serem calculadas as distâncias, vai ser verificado se o ponto do WKT mais próximo da posição do autocarro enviada é superior a um certo limite de metros. Por exemplo, se o limite for definido para 200 metros e a distância entre a posição do autocarro e o ponto do WKT mais próximo for superior a 200 metros, será tido como um desvio de rota. Assim, estes dados serão inseridos na tabela MeanStopTime.

Para os tempos de paragem por paragem, o processamento é o seguinte:

A cada posição do autocarro que é enviada, são obtidas duas paragens: a última paragem pela qual o autocarro passou, e a próxima paragem pela qual o autocarro irá passar. Depois, pegando nas coordenadas da posição do autocarro, será calculada a distância entre a posição do autocarro e as duas paragens anteriormente referidas. Se esta distância for inferior a um limite de metros definido, a informação será colocada na tabela MeanStopTime. Por exemplo, se a distância entre a posição do autocarro e uma das paragens referidas for inferior a 40 metros, esta informação será colocada na base de dados. De realçar que os cálculos dos tempos de paragem apenas são calculados no *BackOffice* da interface web do sistema, quando o utilizador faz o pedido para visualizar esta informação. Neste momento apenas são verificadas as distâncias para se decidir se os dados enviados pela viatura são ou não inseridos na base de dados.

É importante realçar que este tempo de paragem é apenas uma estimativa. A decisão dos cálculos serem realizados num raio de 80 metros em relação à paragem (40 metros antes e 40 metros depois) prende-se com o facto de poderem não haver dados suficientes caso os cálculos fossem feitos apenas com transmissões enviadas pela viatura em que a velocidade fosse zero e com a obrigatoriedade de estar num raio inferior ao estipulado em relação à paragem. Assim, podemos ter situações em que temos um tempo de paragem de alguns segundos, mas em que a viatura na verdade não parou, daí ser importante realçar que trata-se apenas de uma estimativa. Para termos um resultado exato seria necessário que as viaturas enviassem dados a cada segundo, e tal não acontece.

Na figura 4.17 é demonstrado esquematicamente o processo anteriormente descrito, sobre a lógica dos tempos de paragem por paragem.

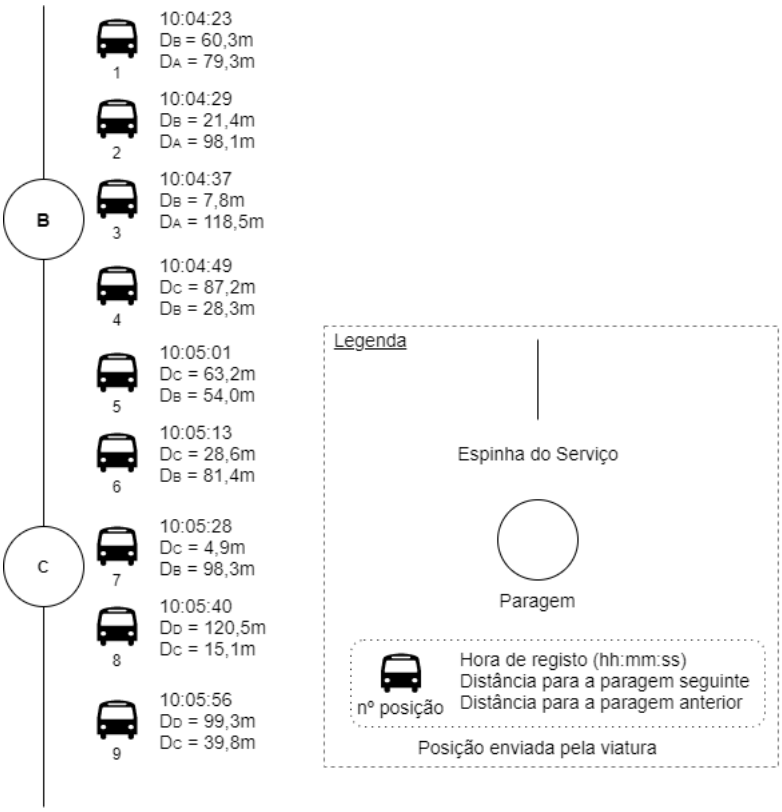

Figura 4.17: Esquema exemplificativo do processo do tempo de paragem por paragem.

Na tabela 4.3 são usados os dados demonstrados na figura anterior. Para cada posição que o autocarro enviou, temos a hora de passagem, qual a paragem anterior e respetiva distância da posição da viatura para esta paragem, e o mesmo para a paragem seguinte. Realçadas estão as distâncias inferiores a 40 metros, o que entra nos parâmetros para que os dados da respetiva posição da viatura sejam colocados na tabela MeanStopTime da base de dados.

Posição	Hora (hh:mm:ss)	Paragem Anterior/Dist. (m)	Paragem Seguinte/Dist. (m)
1	10:04:23	A/79,3	B/60,3
2	10:04:29	A/98,1	**B/21,4**
3	10:04:37	A/118,5	**B/7,8**
4	10:04:49	**B/28,3**	C/87,2
5	10:05:01	B/54,0	C/63,2
6	10:05:13	B/81,4	**C/28,6**
7	10:05:28	B/98,3	**C/15,1**
8	10:05:40	**C/15,1**	D/120,5
9	10:05:56	**C/39,8**	D/99,3

Tabela 4.3: Representação tabular do diagrama da figura 4.17

Terminado todo o processamento dentro da ***MonitorizaçãoAPI***, os dados estão dis-

poníveis para o utilizador visualizá-los através da interface web deste sistema.

Para isso, o utilizador, na respetiva página, terá de selecionar um equipamento, uma data e um intervalo horário para os quais pretende visualizar, caso existam, estas informações aqui referidas. É nesta fase que serão feitos os cálculos do tempo de paragem para cada paragem. Para cada paragem, o tempo de paragem será definido como a diferença de tempo entre a última entrada na base de dados e a primeira entrada na base de dados para aquela paragem, para aquele equipamento em questão.

Utilizando os dados da tabela 4.3, os resultados para o tempo de paragem das paragens B e C são os seguintes:

$\Delta_B = t_4 - t_2 = 10:04:49 - 10:04:29 = 20s$

$\Delta_C = t_9 - t_6 = 10:05:56 - 10:05:13 = 43s$

É possível verificar que a estimativa de tempo de paragem para a paragem B é de 20 segundos, e para a paragem C é de 43 segundos.

Terminados estes cálculos, todos os dados referentes aos limites de velocidade excedidos, desvios de rota e tempos de paragem por paragem estão prontos a serem processados para a forma como o utilizador visualizá-los-á. Para isso, todos estes dados são serializados através da biblioteca *JsonConvert*, de forma a ficar tudo num só objeto de formato string JSON. Este objeto é depois desserializado numa função javascript onde serão criados os pontos para serem apresentados no mapa, bem como o tratamento dos dados para serem exibidos num *pop-up* que é exibido quando o utilizador clica num destes pontos.

Para isto é utilizada a biblioteca javascript chamada *OpenLayers*, que permite representar informações num mapa dinâmico.

O objeto desserializado será convertido em diversos pontos a serem representados no mapa, cada um referente a um tipo de alerta. Cada ponto é um objeto vetorial do tipo OpenLayers.Feature, que irá ter 2 conjuntos de dados:

1. Um objeto do tipo OpenLayers.Geomtry.Point, que contém as coordenadas do ponto a ser representado no mapa;

2. Um conjunto de dados, onde estão definidos o tipo de alerta e outras informações do respetivo alerta (velocidade no caso de se tratar de um limite de velocidade excedido, a distância no caso de se tratar de um desvio de rota, etc.).

Cada ponto terá um estilo semelhante: um círculo, cuja cor irá variar consoante o tipo de alerta (limite de velocidade excedido - vermelho; desvio de rota - amarelo; tempo de paragem - azul). Dentro de cada círculo é também colocada uma letra para ajudar o utilizador a identificar qual o tipo de alerta a que esse ponto se refere (limite de velocidade excedido - ”V”; desvio de rota - ”D”; tempo de paragem - ”M”). Na figura 4.18 são representados uma série de alertas no mapa, após escolhido o equipamento, o dia e o intervalo horário: 5 alertas de limite de velocidade excedido, 3 alertas de desvio de rota

e 2 de tempo de paragem. De notar que os alertas de limite de velocidade excedida e desvio de rota são colocados no local exato de onde a viatura enviou a sua posição. Já os pontos referentes ao tempo de paragem por paragem são colocados na localização de cada paragem, e não nos locais de onde a viatura enviou as suas posições.

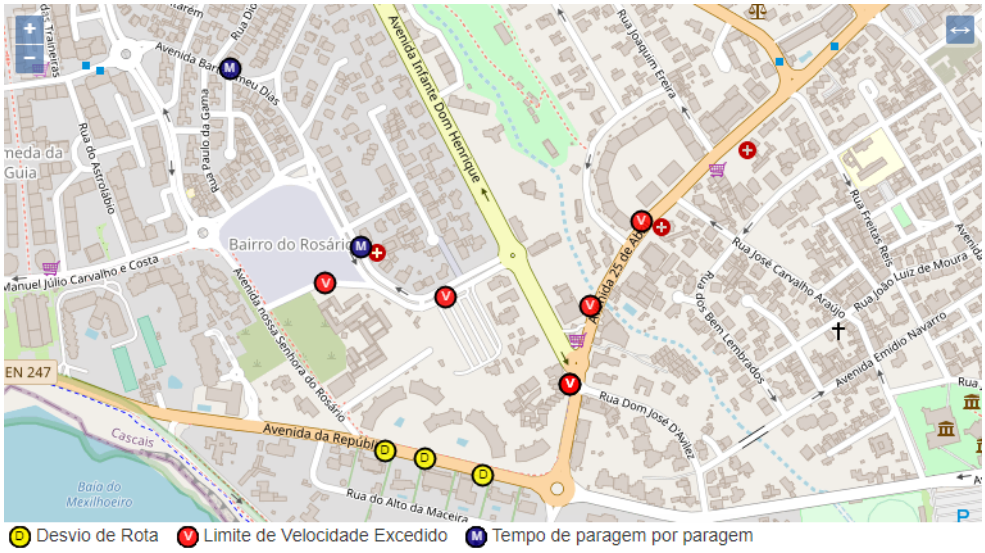

Figura 4.18: Representação de diversos alertas no mapa

As informações sobre cada alerta são depois convertidas para uma string em formato HTML que, o utilizador ao clicar num ponto, serão apresentadas sob a forma de um *pop-up*.

Nas figuras 4.19 a 4.21 são representados de forma genérica os *pop-ups* para cada um dos 3 tipos de alertas.

Figura 4.19: Alerta de desvio de rota.

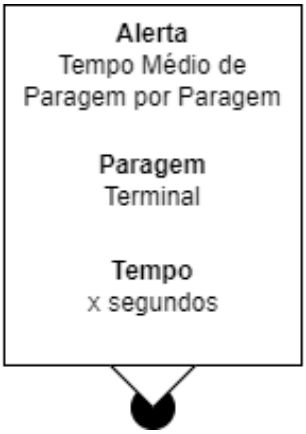

Figura 4.20: Alerta de tempo de paragem.

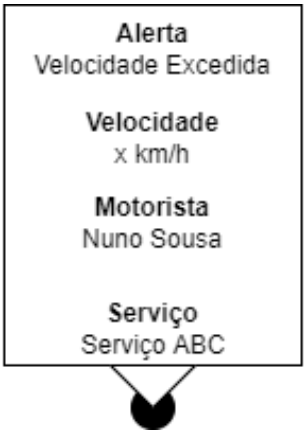

Figura 4.21: Alerta de limite de velocidade.

Por fim, utilizando a informação presente na tabela MeanStopTime na base de dados, foi criada uma nova página no menu gerir serviços onde é apresentada uma tabela com

os tempos médios de paragem por paragem. De referir que apenas são apresentados os dados referentes ao operador que está a efetuar esta operação. Na tabela são dispostas as informações das paragens (nome e código), e de seguida é apresentado o tempo médio de paragem da paragem em questão, em segundos. O raciocínio aplicado ao cálculo destes tempos médios é igual ao efetuado no ponto anterior em relação ao mapa, mas enquanto nesse caso apenas tínhamos um serviço, neste caso são todos os serviços que passaram por aquela paragem, pelo respetivo operador, daí ser o tempo **médio** de paragem por paragem.

Nome da Paragem	Código da Paragem	Tempo Médio de Paragem (s)
Cascaishopping	12345	23
Malveira da Serra	23456	12
Alcabideche (Largo)	34567	39

Tabela 4.4: Exemplo de tabela onde são mostrados os tempos médios de paragem para cada paragem (dados ilustrativos).

4.5.2 *AppMotorista*

Na aplicação ***AppMotorista*** foi adicionada a funcionalidade que permite ao motorista chamar uma aplicação externa, neste caso o Google Maps, para obter uma visualização do percurso a realizar no serviço que iniciou (navegação *turn-by-turn*).

Em primeiro lugar, foi necessário adicionar um novo ícone ao ecrã que o motorista visualiza quando está a executar um serviço.

Depois foi necessário construir o percurso com base nas coordenadas das paragens do serviço em questão. Para isto procedeu-se à verificação do conteúdo de um URL do Google Maps que permita a navegação *turn-by-turn*. Posto isto, a construção do URL é realizada da seguinte maneira:

1. Para cada paragem do serviço, são obtidas as suas coordenadas. Para cada paragem, são colocadas num ArrayList as coordenadas da paragem, no formato "*latitude,longitude*";
2. É criada uma string onde é colocado o prefixo do URL "*https://www.google.pt/maps/dir/*";
3. Para cada coordenada presente no ArrayList referido no ponto 1, esta coordenada é adicionada ao URL, seguida de "/";
4. Sem mais coordenadas para adicionar, é adicionado ao URL um "@"seguido das coordenadas da primeira paragem do serviço. Isto serve para quando o motorista tem acesso à navegação *turn-by-turn*, o ecrã estar centrado na paragem inicial.

5. Por fim, é adicionado ao URL o sufixo "*,13z/data=!4m2!4m1!3e0?hl=pt-BR &authuser=0*".

De seguida está o exemplo do URL de um serviço com 4 paragens (inicial, duas intermédias e final):

https://www.google.pt/maps/dir/38.73951,-9.399765/38.73256,-9.410219/38.72629,-9.411886/38.71841,-9.409042/@38.73951,-9.399765,13z/data=!4m2!4m1!3e0?hl=pt-BR &authuser=0

Nas figuras 4.22 e 4.23 é possível visualizar o trajeto do percurso na aplicação Google Maps, após clicar no ícone para esse efeito.

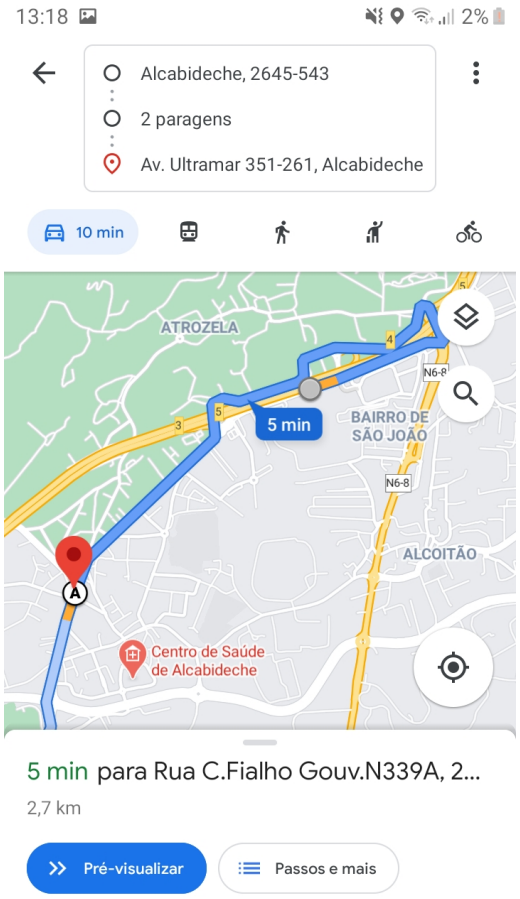

Figura 4.22: Visualização do trajeto no Google Maps com nível de zoom original.

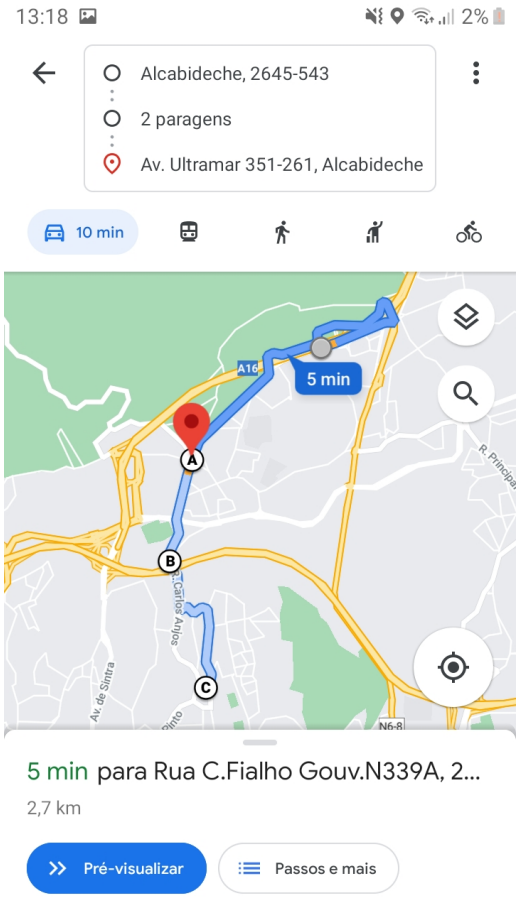

Figura 4.23: Visualização do trajeto no Google Maps após reduzir nível de zoom.

Nas imagens anteriores a paragem inicial está repreentada como um círculo cinzento, sendo que o ícone vermelho presente no mapa refere-se à paragem seguinte. A primeira imagem é o que o motorista vê após clicar no botão de navegação *turn-by-turn* como

representado na figura 4.22. Na figura 4.23 é possível ver todo o percurso do serviço após ter-se reduzido o nível de zoom na aplicação.

Capítulo 5

Conclusão

5.1 Sumário

Este projeto, que tem como tema ”Planeamento de Transporte a Pedido: Gestão da Procura”, teve grande parte do seu trabalho realizado sobretudo em 2 sistemas: o ***Planeamento*** e o ***Monitorização*** e as respetivas APIs. Foi também feito algum trabalho significativo na *api* do sistema ***AppMotorista***. Já na ***AppMotorista***, apesar do trabalho realizado não ter sido em grande quantidade, foi relevante para todo o processo dos serviços ocasionais desde o seu planeamento até à sua execução.

Ainda antes de serem executadas as tarefas planeadas, foi necessário pesquisar e aprofundar a teoria que serve como base a este projeto, nomeadamente os temas de transporte a pedido (*transport-on-demand*) e de mobilidade como um serviço (*mobility-as-a-service*). Esta pesquisa, além de ter como objetivo a aprendizagem do que cada um dos temas abordados, teve também em vista o estudo das soluções já existentes e os possíveis resultados que dessas soluções resultaram.

Sumariamente, todos os objetivos que estavam estabelecidos para este trabalho de projeto foram cumpridos exceto o último, que se referia ao estudo e implementação de uma nova versão do sistema ***Planeamento*** com base em linguagens *low-code*. Isto aconteceu devido a diversos desafios e adversidades que foram sendo encontradas ao longo da execução deste projeto, tais como uma aprendizagem mais prolongada sobre as ferramentas utilizadas, as linguagens de programação e a organização dos sistemas sobre os quais o trabalho iria ser realizado. Também há que enaltecer a adaptação ao mundo empresarial, uma vez que este trabalho de projeto foi realizado numa empresa. E por fim a adaptação ao teletrabalho a partir de meados de março devido à pandemia da COVID-19, onde a adaptação teve de ser feita não só pela mudança do ambiente em que o trabalho era realizado, bem como a dificuldade acrescida de passar a comunicação entre toda a equipa para as plataformas online.

Em relação aos outros objetivos, todos foram cumpridos com sucesso. Desde a criação do percurso a partir do momento em que um serviço ocasional é criado no ***Planeamento***

até à sua execução na ***AppMotorista*** e o respetivo envio de informações no sentido inverso, até à agregação de serviços ocasionais. Algumas das tarefas realizadas não tiverem como base o dito ”transporte a pedido”, uma vez que esta é uma solução que ainda está em desenvolvimento na empresa, e como tal algumas das implementações foram feitas com base em dados de serviços regulares. Porém, estas implementações têm em vista serem aplicadas, no futuro, nos serviços ocasionais, nomeadamente algumas das funções de apoio à decisão que foram implementadas na secção 4.5 do capítulo 4.

No desenrolar do trabalho algumas tarefas foram sendo adiadas e/ou trocadas por outras, algo que é relativamente normal num mundo que está em constante mudança e evolução. Os prazos também nem sempre foram cumpridos devido a alguns atrasos originados pelas diversas razões mencionadas no parágrafo anterior.

Uma vez que o tema do transporte a pedido é ainda uma solução em desenvolvimento por parte da empresa, creio que o trabalho desenvolvido foi bastante importante para o crescimento de um conceito que em Portugal ainda está a dar os primeiros passos.

5.2 Trabalho Futuro

Uma vez que este é um tema ainda relativamente recente em Portugal, e que esta é uma solução ainda em desenvolvimento na empresa, fica claro que haverá certamente trabalho a ser feito no futuro. Quando surgirem os primeiros clientes, possivelmente estes irão colocar novos desafios a serem implementados nesta solução.

Algo que será bastante vantajoso ser implementado no futuro será a opção de um motorista aceitar um serviço. Esta ideia da aceitação de um serviço teria por base o momento a partir do qual o motorista vê os serviços disponíveis a serem realizados, este teria a opção de escolher um desses serviços e aceitá-lo, ficando reservado para ele, tal como funcionam atualmente os serviços de transporte individual e remunerado de passageiros em veículos descaracterizados a partir de plataforma eletrónica (TVDE), como é o caso da Uber ou da Bolt.

Em relação à funcionalidade de agregação de serviços ocasionais, esta também poderá vir a ser melhorada. Poderão vir a ser testados novos algoritmos para a agregação de serviços ocasionais e serem comparados os seus resultados, quer em termos de eficiência no percurso final, quer em termos de ser apresentada uma solução em tempo útil.

Julgo que algo que poderá também ser implementado é dar ao utilizador a possibilidade deste fazer a reserva de um serviço ocasional através de uma aplicação móvel, como por exemplo a já existente ***AppCliente***.

Tal como em praticamente todos os sistemas informáticos que lidam com uma grande diversidade de clientes, terão de haver sempre constantes alterações e os sistemas irão evoluir consoante as necessidades que os clientes vão colocando à empresa. Assim, o trabalho futuro será algo constante, uma solução que irá crescer e será aprimorada à medida

que esta for utilizada.

Bibliografia

[1] Atte Riihelä Jukka Räsänen Ian Sacs Ari Sirkiä Andre Uteng Ari Hartikainen, Jukka-Pekka Pitkänen. Whimpact. 2019. https://ramboll.com/-/media/files/rfi/publications/Ramboll_whimpact-2019.pdf Accessed: 2019-12-07.

[2] Card4B. About us. https://www.card4b.pt/pt/about.html. Accessed: 2019-12-05.

[3] Fundação Francisco Manuel dos Santos. Projecções 2030 e o futuro. 2018. https://www.ffms.pt/publicacoes/detalhe/1586/projeccoes-2030 Accessed: 2019-12-05.

[4] Finanças e Ambiente e Transição Energética Gabinetes dos Secretários de Estado do Orçamento e Adjunto e da Mobilidade. Despacho n.º 1234-a/2019. https://dre.pt/home/-/dre/119190179/details/maximized. Accessed: 2019-12-07.

[5] Adam Frost. New maas travel service launches in stockholm. *Traffic Technology Today*, 2019. https://www.traffictechnologytoday.com/news/mobility-as-a-service/new-maas-travel-service-launches-in-stockholm.html Accessed: 2019-12-07.

[6] Helsinki Regional Transport Authority (HSL). Kutsuplus – final report. 2016. https://www.hsl.fi/sites/default/files/uploads/8_2016_kutsuplus_finalreport_english.pdf Accessed: 2019-12-06.

[7] Médio Tejo Comunidade Intermunicipal. Projeto. https://mediotejo.pt/index.php/projeto. Accessed: 2019-12-06.

[8] Helena Strömberg MariAnne Karlsson, Jana Sochor. Developing the 'service' in mobility as a service: experiences from a field trial of an innovative travel brokerage. 2016. https://core.ac.uk/display/70617333 Accessed: 2019-12-07.

[9] Filipa Almeida Mendes. Mais de 95% da população mundial está exposta à poluição atmosférica. *Público*, 2018. https://www.publico.pt/2018/04/17/ciencia/noticia/mais-de-95-da-populacao-mundial-respira-ar-poluido-1810709 Accessed: 2019-12-05.

[10] PORDATA. Densidade populacional. https://www.pordata.pt/DB/Municipios/Ambiente+de+Consulta/Tabela. Accessed: 2019-12-05.

[11] PORDATA. Veículos rodoviários motorizados em circulação: total e por tipo de combustível. https://www.pordata.pt/Portugal/Ve%c3%adculos+rodovi%c3%a1rios+motorizados+em+circula%c3%a7%c3%a3o+total+e+por+tipo+de+combust%c3%advel-3101. Accessed: 2019-12-05.

[12] Faiz Siddiqui. D.C.-based Split will discontinue rideshare service, citing market 'saturation'. *The Washington Post*, 2016. Accessed: 2019-12-06.

[13] TomTom. Traffic index 2018. https://www.tomtom.com/en_gb/traffic-index/ranking, 2019. Accessed: 2019-12-05.